

# **VAR Grade for Windows**

*Grading Tools for Teachers*

**Version 1.0**

## **MANUAL**

**Written by Dennis Revie**



**© Copyright 1994 by Dennis Revie**

VARed Software  
1490 Calle Fidelidad  
Thousand Oaks, CA 91360-6819  
USA

(805) 523-7546  
CompuServe: 73427,112  
Internet: 73427.112@Compuserve.com

**VAR Grade for Windows** is © COPYRIGHTED 1994 by Dennis Revie.  
All rights are reserved.

**VAR Grade for Windows** is written in Borland International's Borland Pascal. Some of the DLL's are copyrighted by other companies. These are used as per their licenses.

Product names from other manufacturers are mentioned, and are copyrighted and/or registered and/or trademarks of their respective companies. These include Borland's Quattro Pro and Paradox, IBM's OS/2, and Microsoft's MS DOS, MS Word, and MS Windows.

**VAR Grade for Windows** is dedicated to **Wanpen**, whose patience and assistance in producing this program were invaluable.

⇒ **Information**

Information about licensing and registration can be found in the SITELIC.DOC and REGISTER.DOC files. Please read them!

For dealer pricing, volume discounts, site licensing, shipping of product, returns, latest version number, or other technical information, contact us at the address above, or by:

- Phone: (805) 523-7546
- CompuServe: 73427,112
- Internet: 73427.112@Compuserve.com

Last Revision: September, 1994.

# TABLE OF CONTENTS

<b><u>TABLE OF CONTENTS</u></b>	<b><u>3</u></b>
<b><u>I. INTRODUCTION</u></b>	<b><u>6</u></b>
A. Conventions Used in this Manual	6
B. Sample Classes	6
C. Requirements	7
D. Capability of the Program	7
<b><u>II. INSTALLATION</u></b>	<b><u>9</u></b>
A. Files	9
B. Installation With INSTALL	9
C. Installation Without INSTALL	10
D. Networks	11
<b><u>III. RUNNING THE PROGRAM</u></b>	<b><u>12</u></b>
A. Starting the program	12
B. Selecting a Class	13
C. Setting Up a New Class	14
D. Using old definitions for new classes	16
E. Starting a new grading period	16
F. Help, hints, and tutorials	16
<b><u>IV. ENTERING STUDENTS</u></b>	<b><u>18</u></b>
A. General Methods	18
B. Names and IDs	18
C. Typing in Names and IDs	19
D. Importing Students from Files	21
E. Sorting Students	30
F. Transferring Names to a New Class	30
<b><u>V. DEFINING TASKS</u></b>	<b><u>33</u></b>
A. Regular Tasks	33
B. Special Tasks	36
C. View definitions	47
D. Manipulating Tasks	47
E. Examples of Task Definitions	51

<b><u>VI. DATABASE ITEMS</u></b>	<b><u>54</u></b>
A. Defining a New Item	54
B. Altering Database Definitions	58
C. Database Operations	60
<b><u>VII. ENTERING DATA</u></b>	<b><u>62</u></b>
A. Entering Data	62
B. Saving Entered Data to Disk	71
<b><u>VIII. ATTENDANCE</u></b>	<b><u>73</u></b>
A. Date formats and names	73
B. Defining Attendance Dates	74
C. Entering Class Attendance	75
D. Seating Charts	76
E. Attendance options	78
F. Attendance Task Values	79
G. Exporting attendance	79
<b><u>IX. ANALYZING THE GRADES</u></b>	<b><u>80</u></b>
A. Histograms and pie charts	80
B. Correlating Different Tasks	84
C. Plotting the Scores of an Individual	88
D. Class and Student Statistics	92
E. Calculating and Displaying Statistics	95
F. Scaling the Tasks	97
G. Adjusting the Statistics of a Task	98
H. Selecting students to view	98
I. Grade Options	100
<b><u>X. ASSIGNING THE GRADES</u></b>	<b><u>102</u></b>
A. Changing the Grade Names and Values	102
B. Assigning Grade Cutoffs	105
C. What Ifs	106
D. Assigning Final Grades	107
E. Adjusting Grades	108
<b><u>XI. PRINTING AND REPORTS</u></b>	<b><u>110</u></b>

<b>A. Printing header information</b>	<b>110</b>
<b>B. Printing the Student Scores and Attendance</b>	<b>111</b>
<b>C. Write options</b>	<b>119</b>
<b>D. Exporting Data</b>	<b>122</b>
<b>E. Reports</b>	<b>126</b>
<b><u>XII. OTHER PROGRAM FEATURES</u></b>	<b><u>147</u></b>
<b>A. Class Maintenance</b>	<b>147</b>
<b>B. International Support</b>	<b>150</b>
<b>C. Program Parameters--Defaults and Limits</b>	<b>151</b>
<b>D. Beeps or Noises</b>	<b>152</b>
<b><u>GLOSSARY</u></b>	<b><u>153</u></b>

# I. INTRODUCTION

**VAR Grade for Windows (VGW)** is a full-featured Windows program. It helps you to both grade your classes and also to record and quantify attendance in your classes. This is a complete grading system, and will do almost anything that you need to do. In particular, **VGW** will allow you to record, print, and analyze grades and attendance.

There are numerous features in this program, but, in general, **VGW** program allows an almost unlimited number of students, grades, attendance, database items, reports and analyses. You can do statistical analyses and plots, manipulate and print the data in a myriad of ways, and use about any grading system. There are as many grading systems as there are teachers. In that light, the program allows you to alter how it works, both your input and the program output. **VGW** is the *only* grading program that doesn't constrain you to use any particular grading system or method, the *only* one that lets you print data any way you want, and is easily the most flexible and complete system available.

Not only should **VGW** be used for grading your classes, but you should use it's statistical and plotting capabilities to try to understand more about your students and how you evaluate their performance. For example, you can compare scores on two or more exams or homework assignments to see if they produced similar results, or if students did significantly differently on them. This might tell you that one assignment or exam was poorly designed, or that you didn't cover the material adequately. On the other hand, it might tell you that students didn't study for one exam, or didn't take the assignment seriously. In other words, **VGW** is more than just a grading tool, but also an analytical tool that can improve your teaching.

This manual will give you information about **VAR Grade for Windows**. If this is the first time you have tried out the program, you should try the tutorials in the program by selecting "Help Tutorial". They will teach you how to use the most commonly used features of the program. The plentiful features of the program can sometimes intimidate the new user, and the tutorials are designed to speed up learning the program. In addition, there are hints that appear each time you use the program. These are designed to increase your familiarity with the rest of the program: that part you aren't currently using. This manual details all of the features of the program. The purpose of this manual is to explain those features that are not obvious to you when running the program.

## A. Conventions Used in this Manual

For this manual, we assume that you have a basic familiarity with Windows. You know how to use a mouse, know how to start a program, and are familiar with how icons, menus, and buttons work.

- Names of files and programs are printed in UPPERCASE letters.
- Warnings and notes are printed in *underlined italics*.
- Examples of what to type are in double quotes: "TYPE THIS".
- MS-DOS prompts are usually listed as: A>. Anything after the prompt should be typed.

Example: A> GRADE

The prompt is A>, and you should type "GRADE".

- Special keys are those that do not print normal (ASCII) keys. They are designated in the manual and program by bracketing the key name with "<" and ">".

Example: <Esc> means the Escape key, <Rtn> and <Enter> mean the Return or Enter key.

- When the manual describes a procedure and states "select", you can use either a mouse or keyboard to pick a menu item or button. Some items, like the "Help Tutorial" mentioned above, contain two or more words. The first word is usually used to select the first menu item, and the rest selects an item on the submenu. "Help Tutorial" means to select "Help" on the Main Menu, then "Tutorial" on the submenu. The underlined letters are menu items that can be selected by the keyboard. Buttons described in the manual do not have an underlined character.

## B. Sample Classes

To look at the capabilities of **VAR Grade for Windows**, four sample classes, called SampleA, SampleB, SampleC, and SampleD, have been included with the program. Many of the features of the program have been used to set up these classes. To view them when starting the program, select "File Open".

## C. Requirements

**VGW** requires MS Windows 3.1 or an operating system that can run Windows 3.1 programs (like OS/2). 4 Megabytes of RAM (random access memory) are required, as well as 4 Megabytes of disk space on your hard disk. In other words, there are no special requirements except for Windows 3.1.



## **D. Capability of the Program**

This program allows you to have up:

- 16,000 students that can be separated into up to one section per student
- 200 database items per class
- 2,000 tasks per class
- 240 attendance days per class and 2 seating charts per section of students
- an unlimited number of classes.

In reality, the limitations are due to available memory on your computer and hard disk.

The program runs on networks, but does not use any special capabilities of networks.

Custom versions of this program are also available. When you register your current version, should you desire a custom version, indicate so and we will try to meet your requirements. Some changes can be done with little effort, and can be done for free. Some changes cause significant effort, which would require additional payment. We are also always happy to have you suggest improvements for the next update of the program.

## II. INSTALLATION

**VAR Grade for Windows** is installed by running the program INSTALL. If you prefer to install the program yourself, directions are also given below. This chapter gives information on using the INSTALL program to install **VAR Grade for Windows** and information on installing the program yourself.

The files included with the program are listed in the README.WRI file. This file also includes general information about the program.

### A. Files

For the program to run, it requires:

- ◆ VGW.EXE (program)
- ◆ DEWCCVG.DLL and DEWTCVG.DLL (required DLL files)
- ◆ GSW.EXE and GSWDLL.DLL (Graphics Server and DLL)
- ◆ BWCC.DLL (Borland Windows Control DLL)

These files are used by the help and tutorial systems:

- ◆ VGW.HLP (help file)
- ◆ VGW.TUT (tutorial)

These files are created when the program runs:

- ◆ VGW.INI (global defaults)
- ◆ VGW.INT (translations)

Other files of interest:

- ◆ README.WRI (read me file)
- ◆ README.DOC (same as README.WRI, except ASCII)
- ◆ REGISTER.WRI (information on registration)
- ◆ SITELIC.WRI (information on site licenses)

There may be other files, as well. See README.WRI for details. The files will be in a ZIP file called VGW10.ZIP.

### B. Installation With INSTALL

To install the program, you run the program **INSTALL**. Follow the instructions below. You need 4 megabytes of disk space on a hard drive.

1. Put the disk into a drive. The disk can be in any disk drive. For the instructions below, it is assumed that the disk is in drive A. If you use drive B, or any other drive, replace the "A:" in the instructions with "B:" (or "d:", where "d" is the letter of the drive).
2. Select "File" on the menu in the Program Manager, then "Run" to run **INSTALL**. Then type "A:INSTALL". If the **VAR Grade for Windows** installation disk is in another drive, type "d:INSTALL", where "d" is the letter for the drive.
3. An opening screen will appear, explaining the hardware requirements of **VAR Grade for Windows**, and ask for the installation directory. The suggested directory is "d:\VGW", where "d" is the drive letter. You can change the install directory, if desired.
4. The program will then install the program. Most of the files will go into a subdirectory of the one you chose, while the **README** and sample classes will go into that directory. The program will be installed into a new file folder. If **INSTALL** did not find the **ZIP** files it needs, it will ask for the new directory.
5. You will then be shown the **README** file.

### **C. Installation Without **INSTALL****

1. You need up to 4 megabytes of disk space and the program **PKUNZIP 2.0** (if not distributed with the program, it is available from disk vendors and BBS systems).
2. The files you need for the program are listed in Section A.
3. Make a new directory, for example, \VGW. Move to that directory. If the disk with the program is in drive A, type "PKUNZIP A:VGW10".
4. You need to make a new program group and program item to run **VGW** from Windows.
  - To add a new program group, select "Files New", then "Program Group" and then "OK". Next, enter a description of the program, e.g., "VAR Grade for Windows 1.0". Finally, move to the Program File line and enter "VGW.GRP". You now have a program group.
  - To add the program, select "Files New", then "OK". Enter a filename of "\VGW\

VGW.EXE” (assuming the directory for **VGW** is \VGW). You can also add a program description (“Great Grading program”), and then “OK”. The **VGW** icon should appear in the program group.

5. If you want to start the program with drag and dropping, you need to add two lines to a Windows file. It is important that you do not change anything else in this file, as all Windows programs use the file. This is described below.

- Start Windows Write.
- Select “File Open”.
- Enter “win.ini”. When Write asks whether to convert the file, select “No Conversion”.
- Find the line that says [Extensions].
- *Add* the following two lines anywhere in that section of the file:

```
PAR=C:\VGW\VGW.EXE ^.PAR  
PAX=C:\VGW\VGW.EXE ^.PAX
```

- Select “File Exit”.
- When you are asked to save the changes, select “Yes”.

6. Read the README.WRI file.

## **D. Networks**

This version will run on networks (LANs). It will not, however, take advantage of special features of the networks, such as interactive mail or print queues.

If you have installed the program on a network, you should run the program once and set the program parameters, such as grading systems and translations of output terms, to those most used by your users. The program searches for the VGW.INI and VGW.INT files first in the local directory, then in the program directory. Setting these will prevent you from having to modify the same values again and again.

## III. RUNNING THE PROGRAM

This part of the manual tells you how to start the program, including where to place your class files and how to locate class files from within the program. It will also explain how to access the help from within the program. There is also a long tutorial that explains the basics of how to use the program. To use this part of the manual, **VGW** must be installed (see Chapter II).

There are several ways you can start **VAR Grade for Windows**:

- ◆ Standard way: double click on the program icon.
- ◆ Fancier: make a program icon for each class, and change the File Name on the command line to “\VGW\VGW.EXE class”, where “class” is the name of the class (including the directory).
- ◆ Drag and drop: when in the File Manager, drag a class file (ending in .PAR) to VGW.EXE. That class will be loaded into **VGW**.
- ◆ Associating a class to **VGW**: make an icon for a class, and associate it with **VGW**. Clicking on the icon starts the program with that class.

### A. Starting the program

#### 1. Double clicking

When the program was installed, an icon was made that starts the program. Double clicking on the icon will start the program. The program starts the last class you designated, and sets the window coordinates to those from the previous session. For example, if **VGW** was maximized when last run, it will start maximized.

In the file properties for the icon, you can change the startup directory of **VGW**. By designating a different directory, you can put your class files in a different directory than the program. When the program starts, the directory will be set to that name. If you now select “File Open”, files in that directory are listed.

One method of setting up your classes is to have each class in a different directory. You can make **VGW** start each class in a different directory by copying the original icon, and changing the working directory to the new directory. This will start **VGW** in that directory. An additional improvement on this is to change the program command line to read “\VGW\VGW.EXE class”, where “class” is the name of the

class. Now, when you double click on the icon, the program will start with the desired class instead of the class last viewed.

## 2. Drag and drop

**VGW** can be started by dragging and dropping, as well. The simplest method is to use the File Manager (or a similar program) to select a class file. These files end in .PAR. Hold the left button down, and move the mouse to the file **VGW.EXE**. Now release the button. The File Manager will ask whether to run the program, select “Yes”.

## 3. Association

A different method associates a class file with the program. Make an icon for a class by selecting “Files New”, then “OK”. Enter a filename of “\path\class.par” (where the directory is substituted for \path and class is the name of your class), and then “OK”. The **VGW** icon should appear with the class name substituted for the **VGW** name. Any time you click that icon, the program starts and loads that class.

## B. Selecting a Class

The first time you run **VGW**, no class has previously been loaded. **VGW** will open the “File Open” dialog, and let you select a class. This is the standard Windows dialog, except that only classes are listed. Any other time you start the program, **VGW** will load the last class (unless you associate, drag and drop, or specify a file on the command line: see Section A).

### 1. Opening a class

There are two different ways to open a class.

- First, you can select “File Open” from the Main Menu. All classes in the current directory are listed. You can select the desired class. If you have made changes to the class currently open, **VGW** asks whether to save the current class.
- Second, you can select a file listed at the bottom of the File Menu. The five last classes opened are listed.

### 2. Starting new classes

Select “File New” from the Main Menu. If the current class has been changed, you will

be asked whether to save the current class. When you finish adding data to the class or save the class, you are asked for the class name.

### **3. Recovering a class**

Recovering a class retrieves data from the previous time you changed the class. Every time data is saved, a backup is made of the previous set of data. This is analogous to the “.BAK” files made in word processors.

Select “File Recover” from the Main Menu. A dialog will appear that is the same as the file open dialog, except the class backup files are listed (those ending in “.PAX”). Select the class you want to recover. The class will be read into the program. You need to save the data to recover it, and can either save the class data to a new name by selecting “File Save as”, or overwrite the current class data with “File Save”.

## **C. Setting Up a New Class**

After making a new class, you will want to add data. Although later chapters in this manual fully explain all the types of data and how to add, delete, change, and otherwise alter the data, this section will introduce some of the things you might want to do.

We recommend that the first time you use the program, you look at all the menu options (you don't have to select the items, just pull down the menus). This will give a first look at the program capability.

**VGW** can be used for attendance, seating charts, grades (tasks), and personal data (database items). In addition, you can separate the students into “sections”. Sections are students in one classroom. One common example is that, at large universities and colleges, you will have one large lecture class that has students broken down into smaller groups for discussion or labs. Each lab or discussion group would be a section of students. A second example is a teacher who teaches several identical classes. A high school history teacher may teach four classes of American History. If each class has identical assignments, you can put all the students into one “class” that has four sections of students. This makes it easier to analyze and compare students in different classes.

Other factors that you might want to consider are program colors, the grading system, and whether to use passwords to protect your data. Your name can also be entered and printed on reports.

### **1. Using multiple sections**

If you want to divide your classes into sections, you should select “Option Multiple Sections”. This will allow you to add and analyze students from as many sections as needed. If you don't select this option, only one section of students is used by the



program.

## 2. Colors

The colors of the program can be changed on the “Edit Options” menu. There are two basic choices.

- **VGW** colors: these are the colors you see when you first run the program.
- System colors: these are standard Windows colors selected on the Control Panel. To change colors, you need to run the “Control Panel” program in the Main Windows folder. These colors give you more flexibility, but affect most Windows programs.

## 3. Setting up a grading system

Although some schools don't use letter grades, most do. You can set up a grading system by selecting “Grade Grade names and values” from the Main Menu. You can use the pre-defined grading systems that are listed, or enter your own.

A grading system has three parts. First, there is a grade name (e.g., “A”, “B”, etc.). Second, there is a grade value. The value of an A could be 4.0, a B value might be 3.0, etc. Third, there is a grade cutoff. This is the score needed by a student to achieve that grade. You might need a cutoff of 90 to earn an A, an 80 to earn a B, etc.

*Note:* If you mix letter and number grades, you may want to set the letter grade values to numbers from 0 to 100, rather than the ones the program has predefined. See Chapter X for more details.

## 4. Passwords

If many people have access to your computer and its disks, you may worry about someone's altering your class files. To greatly reduce the chances of this happening, you can require that a password be used for your class. To define a password, select “Option Password” from the Main Menu. Not only can the whole class have a password, but you can also protect each section of students with a different password. If you use passwords, you have three chances to enter the correct password. Entering a class password allows you to see and use all the students in the class. Entering a section password only allows you to see and view that section of students. This allow you to have one master class that you know the password to, and let each teaching assistant only be able to see and alter data for their own section.

When you enter passwords, you get three chances to type in the correct password.

*Note:* The default value is to not have a password. The password can have up to 15 characters, numbers, and/or punctuation characters. Should you forget your password, you can contact us at the address at the beginning of the manual to get instructions on how to defeat the password. The password is not foolproof, but greatly reduces the likelihood of snooping.

## **D. Using old definitions for new classes**

If you want to use the same exams and other class information for a new class, there are three major ways to do this.

- Save the current class with a new name by selecting “File Save as”. Next, delete all the students by selecting “Edit Empty the class”. This will remove all the students while keeping all the task and attendance data intact.
- Export data by selecting “File Export”, then one of the options. You can select task and database items to export. Next, select a file name. Next, select “File New”, and then “File Import” and import that data. Last, delete the students by selecting “Edit Empty the class”. This method is explained in more detail in Chapter IV.
- Export part of the data by selecting “Task Export tasks”, or “Attendance Export dates”. This is explained in more detail in Chapters V and VIII.

## **E. Starting a new grading period**

If you start a new quarter or semester with the same students, you can delete parts of the data to create a new class. Select the old class, then select “File Save as” to save it with a new name. Next, selectively delete what you don’t need. For example, to delete student scores but not the tasks, select “Task Clear scores”. Next, select all the tasks. You can delete selected tasks, attendance dates, and even seating charts. If you want to use some of their old task scores in the new grading period, such as their final grade, just don’t delete that task or clear scores from that task.

## **F. Help, hints, and tutorials**

VGW also has a help system, hints, and tutorials. These should help you learn how to use the program.

### **1. Hints**

When the program starts, a hint screen appears that suggests some actions to take when running the program the first time. Similar screens appear the first time you enter many areas of the program. After you view it the first time, they no longer appear. The purpose is to help guide you through the intricacies of the program. Selecting “Turn off hints” will prevent all of them from showing. To turn the hints back on, select “Help Hints”.

## 2. Tutorials

You can also run tutorials from **VGW**. These are designed to help guide you through specific things you want to accomplish. They can stay on screen while you try to master a new feature. To run a tutorial, select “Help Tutorial” from the Main Menu.

## 3. Help

**VGW** also features a help system. There are several ways you can access and view help:

- Select “Help Index” from the Main Menu. This will put you in the Help Table of Contents.
- Press <F1> anywhere in the program. This will put you in the Help Table of Contents.
- Press <Ctrl F1> from anywhere in the program will show a Help topic related to the current place in the program (“context sensitive help”).
- Press <Shift F1> from anywhere in the program will change the cursor to a “?”. If you select an item from the Main Menu, you will get help on that topic. The “?” appears when the mouse pointer is on the Main Menu or on the main window when nothing appears below it.

## IV. ENTERING STUDENTS

One of the first things to do when starting a new class is to enter the names of students into the program. This chapter tells you how to enter new students into the class, add IDs, change the names of students, and sort and list the students.

### A. General Methods

Each student in the class must have a name. Students do not need an ID, or any other data entered. IDs can be used for anything a student name can be used: sorting, printing, identification of students. If you use IDs but not names, you should enter them into the names column, then change the title of names by selection “Option Output terms”, then “Common terms”.

You can also enter student data such as addresses, gender, and year in school as database items. Student grades are entered in tasks, while attendance is entered as, not surprisingly, attendance.

- You can type the names into the program when entering data for an individual student or when entering class data.
- You can import names into the class from flat files or comma and quote files.
- You can easily enter names and/or IDs in the quick entry section of the program.

### B. Names and IDs

Student names and IDs can have up to 30 characters. However, initially the student names are limited to 20 characters long and IDs are limited to 15 characters. You can change the names to between 5 and 30 characters, and the IDs to from 0 to 30 characters (0 means you don't use IDs). These values are important primarily for entering data, as they affect the width of columns and how many characters can be entered. To change the values, select “Option ID and Name sizes”.

The program can sort the students alphabetically by their names, so it is usually better to enter the names *last* name first. Spelling and capitalization is retained exactly as you enter it, except that leading and trailing spaces are removed. When sorting, however, lower and upper case letters are not distinguished, so “Sue” is considered to be the same as “sue”. When printing, you can print the names either as is, or you can “flip” them. To flip is to reverse the first and last names. “John Smith” would become “Smith, John”. “Johnson,

Anna” would become “Anna Johnson”. However, as a warning, if you flip “John Standing Tree”, you would get “Tree, John Standing”, instead of the correct “Standing Tree, John”. You could get the last example to flip correctly if you entered it “John Standing\_Tree”.

ID numbers or other names (nicknames, etc.) can be entered as “IDs”. At some colleges and universities, students are given their grades by student IDs rather than by name. If you want to separate the first and last names, you can, if desired, use IDs for first or last names and the "name" for the last or first name. If you do not use ID numbers, setting the length to 0 will stop the program from asking you about IDs. Note that database items can be used for any type of data, including nicknames.

The program can sort by IDs, names, any database item, or by scores on any task. When sorting, spelling is retained, but, as in sorting names, sorts do not distinguish between lower and upper case letters. See Section E for more details.

After names, IDs, and section numbers have been entered into **VGW**, you can prevent their alteration. Select “Edit Options”, then check the “Prevent alteration of ...” options.

## **C. Typing in Names and IDs**

### **1. Quick entry**

You can quickly add students by selecting “Edit Quick entry of ...”. There are two options: adding only student names, or adding both names and IDs.

If you use multiple sections of students, you must first pick which section to use for entering the new students. The sections numbers can range up to 65530. To select a section, either type in the new section number, or select a student in the existing section. Then select “OK”.

To enter names, you just need to type in the name, ending with a <Rtn> or with an “Add”. If the name is already present in that section, it is NOT reentered, as only one copy of a name is allowed per section of students. However, the same name could be entered in two different sections of the same class. Names type in can be rejected and then corrected. Note that names can be changed later in several parts of the program (see below), so unnoticed errors can be fixed at a later time.

If you are also entering IDs, pressing a <Rtn> while on the name field moves the cursor to the ID field. After typing in an ID, pressing <Rtn> adds the name to the list on top and the cursor moves back to the name field.

## **2. Class data entry**

Although the quick entry is the easiest method of typing in a lot of names, at times you only need to add a couple of names.

When on the Class Data Entry window, there are two ways to add names. First, if the name field of a row is empty, you just need to move the cursor to that field, then type in the name. Second, if all the rows are filled, you can use a button to add a new field. Selecting the “Add row” button will add a new row at the end of the list of students. Selecting the “Ins row” button will insert a new row at the row of the cursor. The students below that row are moved down one row.

Rows that have empty names will cause the student to be deleted. For example, if row 3 had the name “Crick, Francis”, and you changed it to be “” (blank), then all the data will be removed when the Class Data Entry window is closed or modified. This is one way to remove students from the class. For additional information on changing names, see section 4, below.

After all names have been entered, you can prevent changes in them. Select “Edit Options”. Then select “Prevent alteration of names”. IDs and section numbers can also be protected from alterations. Preventing alteration of names also prevents adding or deleting any students. The “Add row”, “Ins row”, and “Del row” buttons are then grayed and unusable.

## **3. Individual entry**

You can add a single student on the Individual Entry window. This window allows you to change data from one student at a time.

To add a student, select “Insert” from the menu. A new student is inserted in the class. You then enter the name and any other data that you want. An blank name field when moving to a different student will cause the student to be removed from the class.

## **4. Adding, changing, and deleting names, IDs, and sections**

Names, IDs, and section numbers can also be added, changed, or deleted.

### **(a) When editing data**

Names can be altered by entering anything different for the name. For example, if a name is “Watson, James”, you can change it to “Watson, Jim” by moving to the first

name, then deleting the “James” and adding “Jim”. A blank name will delete the student from the class.

NOTE: when a student has been deleted, his or her name and all associated data are deleted.

Changing the ID, or changing the section number for the student, is done in a similar way. Blank Ids are OK. Blank section numbers will not affect the student section. You need to change it to a different number.

If you have prevented alteration of names (see Section 2, above), you are prevented from adding or deleting students. If you prevent alteration of IDs and/or sections, you just cannot change the values when editing.

### **(b) Moving a section of students**

You can move all students from one section into a second one by selecting “Edit Move a section”. You then select the section to move and the new section number. For you to move section number, you must have checked “Multiple sections” on the “Option” menu.

### **(c) Removing groups of students**

You can remove all students from one section by selecting “Edit Remove a section”, or all students from the entire class by selecting “Edit Empty the class”. Be careful when you use these options, as all the data for the students that were removed is lost (although they can frequently be recovered by using the methods in part d, below).

### **(d) Recovering deleted data**

Data that has been changed or deleted is normally lost. There are, however, two ways to recover the data if you immediately decide to “undelete” it:

- Quit the program without saving the data to disk. Any data entered in this session of the program will not be saved.
- If you have already saved the data from the class, use the backup files to recover the data from the previous session. This is simple to do. Select “File Recover”, then the name of the class. Data from the previous session will replace the current class. You then need to save the data by selecting “File Save”.
- Every time you add data, you should write the data to a printer. If the first two



methods fail, you can reenter the data using those printouts. This is obviously not a preferred method, but, at times, may be essential. *We therefore highly recommend printing your data each time you enter new data.*

## D. Importing Students from Files

You can enter names via standard text files (also called ASCII files). **VAR Grade for Windows** will import data that is written by many other programs, including most spreadsheet and database programs.

When data is imported into **VGW**, either a student name or an ID *must* be included for each line. If a student name is included, the program will check whether the name is already present (as well as whether the section is correct, if used). If the name is present, the new data will be included with that student's data. If the name is not present, the program will add the data as a new student unless told not to do so (see below).

If you do not include a name on each line, the program will check the ID to see if it matches a student in the class. If so, the new data will be added. Otherwise, the data is discarded: you cannot add students only by ID, as a name is required. If a student name and ID is included, the name only will be checked. If the student exists, the old ID will be changed to the new ID. Each line that is imported can have 1024 characters.

For tasks and database items, the program will try to match the task number, task name, or database name. If it matches them as being identical to one already defined, the new data will replace the old data. Otherwise, a new task or database item will be defined and added to the class.

You can prevent the program from adding new students when importing data by selecting "File Import Options", then selecting "Don't add students". This allows you to have data from lots of different classes in one file be imported into the program. Only the data for students in the correct class will be imported.

### 1. Importing flat files

Many spreadsheet and database programs can export (write) data into a flat file. A flat file is one where the data is listed in columns. All the names of students are in one column, all the IDs in another, etc. To import data from a flat file, select "File Import Flat file". Flat files can be used for importing of student names, IDs, sections, database items, and/or task scores.

You tell **VGW** which columns the name, IDs, database items, sections, and tasks are in,

and the data will be automatically loaded into the class by including a header line. Header lines that do not have student names should be deleted, as the program will try to load those lines as well. An example of a flat file is listed below:

Jones, Fred	123	1	34.5	23.5	B	44
Smith, Ann	321	2	43.5	18.3	B	43

To load new students into the class, go through the following steps:

- \* The program first asks for the filename of the flat file. You can pick the file from a list, or move to another directory to pick the file.
- \* The program then asks what type of data is in the file. Check all the boxes that are to be imported. Make sure that either the name or ID box is checked.
- \* If you checked tasks or database items, you need to specify which tasks or items are in the file. The database items are first selected, then the tasks.
- \* You then select the order of the items in the file. If you don't want to import a field, leave it on the upper list.
- \* The program then displays the first few lines of the file across the top of the screen in a box. If necessary, you can scroll the box left and right. You need to tell the program where the data is located by clicking the mouse on the first and last columns of each piece of data. You start over or back up by pressing the appropriate buttons.
- \* Data is then read into the program. If the task data is numeric, it will be read as a number task; if it is not numeric, the data will be read as a letter task.

In the above example, suppose the first column is names, the second IDs, the third sections, and the next four are tasks. The sixth column (third task) would be read in as a letter task. The tasks would be numbered as the first four unused tasks in your class, and given names (e.g., Number task 1, Number task 2, Letter task 3, and Number task 4).

## 2. Automatic flat file imports

Automatic flat files import data from flat files, as described in section 1. However, the program doesn't ask for where the columns are located or what data is in each column. Instead, **VGW** tries to determine what is in each column and where the columns are located.

To automatically import flat files, select “File Import Automatic flat files”. You are asked for the file, and whether the data is separated by spaces or by tabs. Automatic imports work well with tab separated data, but not as well with data separated only by spaces. First and last names are usually separated by a space. This makes determining where the columns are very difficult.

You can force the data to be a specific type by adding a header line. The method of determining the type of data is the same as that used for the comma and quote files, and is described below in that section.

Because of the problem with data separated by spaces, it is better to import them with the method described in Section 1. If you have a choice, either comma and quote files or flat files separated by tabs are the best way to import data, not data separated just by spaces. Note that most spreadsheet and database programs can export data as comma and quote files or as tab separated (delimited) data.

### 3. Importing comma & quote files

Many spreadsheet and database programs also can read in data from files where the data is not in columns, but is instead separated by commas. Data that has commas in it, e.g., student names, are kept together by double quotes (“Jones, Paul”). An example of data in the file might be:

```
"Jones, Fred", "123", 1, 34.5, 23.5, B, 44, "Junior"  
"Smith, Ann", "321", 2, 43.5, 18.3, B, 43, "Senior"
```

To import a comma and quote file, select “File Import Comma and quote”. You are then asked for the file name. The data is then read into the program. To determine what data is where in the file, the first line is checked. The following two sections describe ways that the program determines the data types.

#### (a) Program determination of fields

For the above data, the program would determine that the student names are in the first field, the IDs in the next, the next five are tasks, and the last is a database item. For the tasks, the fourth one will be a letter task; the others will be number tasks. As in the flat files, the names of the tasks will be made by the program, and the first five unused tasks will be used for the new tasks. The program determines this by following a certain set of rules.

- ◆ The first item surrounded by quotes is assumed to be names.
- ◆ If the second item surrounded by quotes is assumed to be IDs.
- ◆ Other items surrounded by quotes are assumed to be DB items.
- ◆ If an item is not surrounded by quotes, it is either a number or a letter task.

If this is not what you want, you have another option. You might include this as the first line of the file:

```
"Name", "ID", "Section", "One", "Two", 8, "#4", "Year"
```

This line would tell the program that the first field is the name, the second is the ID, the third is the section, the fourth and fifth are number tasks (they are numbers) with the names One and Two, the sixth is a Letter task 8, the last task is a number task with the name #4, and the last field is a string database item with the name Year, and a

length of 6 (the length of "Junior", above).

NOTE: If a header line is included, whether items are letter or number tasks, or database items, is determined by the student data on the second line, not from the first line.

- “Name” tells the program that field contains student names.
- “ID” tells the program that field contains IDs.
- “Section” means that field contains section numbers.
- Other words are treated as tasks or database items, depending upon the data on the *second* line.

The three predefined words described above need to be the entire label for the fields, but whether the labels are in upper and/or lower case letters is not important. For the tasks, anything that is a number (like 8) is taken as the desired number of the task (the program will make up a name for the task), and anything that is not a number will be taken as the name (with its number being the first unused task). Whether the tasks are made number or letter tasks are determined by the data on line two (the first student). Data from the second line that fits neither numbers or the allowed values for letter grades are assumed to be database items.

### **(b) Forcing data types**

Since the program can guess incorrectly about which type of data is in each field, you can tell the program exactly which type of data is in each field by using the method outlined below.

"Name", "ID", "Section", "One\n", "Two\n", 8\, "#4\n", "Year\s10"

Replacing the line in part a with the above line will force the definitions of One, Two, and #4 as number tasks, 8 is a letter task, and Year is a string type database item of length 10. The backslash (“\”) at the end of the data type on the first line, then a letter, tells the program what data type the item is. The list of data types is below.

#### Command

\a??

#### Definition

Attendance item. ?? is any number from 1 to 240.

\l??

Letter grade. ?? is any number from 1 to 10000.  
The number is the task number of the data.

\n??

Number grade. ?? is any number from 1 to 10000. The number is the task number of the data.

<u>Command</u>	<u>Definition (database items)</u>
\b	Boolean database item.
\c	Character database item. Any characters are accepted when reading in the data. If you read in data with a Character type, it is suggested that you "redefine" the item to pick allowed characters. Otherwise, you may not be able to change the data.
\d	Date item
\i	Integer database item (numbers from -2,147,483,647 to +2,147,483,647).
\s??	String database item. ?? is any number from 1 to 255. If the length is over 40, it is called a Memo item.
\w	Word database item (numbers from 0 to 65535).

Note: Word and Integer from **VAR Grade** have been replaced by the new Integers, and are also called Integers.

Examples:

- ◇ In a comma and quote file, the item "Telephone\s13" will set the item to be a String database item, of length 13.
- ◇ "Grade in school|i" will set the item to be an Integer database item.

#### 4. Entering scores of one task

Importing files of one task is designed to be a simpler import of grades from a flat file that contains one task and one type of identifying data (name, ID, or database item). The main purpose of this option is for instructors in large classes. They can easily import grades on an assignment from their teaching assistants. Either names, ID's, or a database field are matched with students in the class, then the new grade will *replace* the previous one. The files imported need to have a name, ID, or database item listed first, then a tab or spaces, then the task score.

The method works like the flat files import method. When you enter this part of the program, you are asked for the file name, the task number, and whether to match names

or ID's. If the teaching assistants use **VGW**, it is easier to export and import grades directly by selecting "File Export Directly to another class".

The file format should have names or ID's like the following example (listing ID's, then the task score), and there should be no first line header, as you are asked for the details before importing the data:

```
A1234567 66  
Z6543210 44
```

## 5. Importing data from Paradox for Windows

Described here is how to export data from Paradox in a format that **VAR Grade for Windows** can understand. Because comma and quote files work best, they are used for the example.

- Start Paradox for Windows. If you want to export only part of the table, you will need to make a new temporary table that contains the data you want to export. Design a query that selects the data you want.
- Select "File Utilities Export".
- Select the table you wish to export on the left list box, and "Delimited text" on the right list box.
- Specify a file name for exporting. Paradox asks for a filename, type in the name. *Write down the name, including the directory.* You may also wish to check the options. The export should be set to "Comma", "Quote", and "Text fields only".

Paradox for Windows now exports the table to a comma and quote file.

Paradox for Windows does not export the names of the columns. You probably want to name them before importing them into **VGW**. Make an additional line at the top of the file that contains the data types described above. Student names should have the header "Name", IDs should be named "ID", and section numbers should be titled "Section". For **VGW** to locate the correct student, the spelling must be the same as the student name or ID of the existing student.

To import the file into **VGW**, follow the directions in importing comma and quote files (Section 3). It is a good idea to check the data that was imported *before* saving it to see if any problems occurred in the export or import.



## 6. Importing data from Quattro Pro for Windows

Described here is how to export data from Quattro Pro for Windows (QPW) in a format that **VAR Grade for Windows** can understand. Because QPW only exports tab separated (delimited ) flat files, they are used for the example.

- Start QPW, and the spreadsheet that has your data.
- Select the spreadsheet page you want to export. You may have to adjust the cell widths and page layout before writing the data.
- Select “File Save as”. Type in the name, with an extension of “.TXT”. The correct extension is important. *Write down the file name, including the directory.*

QPW now exports the data to a tab delimited flat file.

To import the file into **VAR Grade for Windows**, select “File Import Flat files”. Then enter the name of the file you wrote down, above. Follow the directions in Importing flat files (Section 1). It is a good idea to check the data that was imported to see if any problems occurred.

An alternative for importing the data is to alter the first line of the spreadsheet page to use as a header and explicitly specify the data types. Then automatically import the data (see Section 2).

## 7. Importing data from Microsoft Works

Described here is how to export data from Microsoft Works in a format that **VAR Grade for Windows** can understand. Because Works exports tab separated (delimited ) flat files, they are used for the example.

- Start Microsoft Works, and the spreadsheet or database that has your data.
- Select “File Save as”. Type in the name, with an extension of “.TXT”, and select a file type of “Text and Tabs (DOS)”. The correct file type is important. *Write down the file name, including the directory.*

Works now exports the data to a tab delimited flat file.

To import the file into **VAR Grade for Windows**, select “File Import Flat files”. Then

enter the name of the file you wrote down, above. Follow the directions in Importing flat files (Section 1), or Automatic flat files (Section 2). It is a good idea to check the data that was imported to see if any problems occurred.

An alternative for importing the data is to alter the first line of the spreadsheet page to use as a header and explicitly specify the data types as described in Section 3. Then automatically import the data (see Section 2).

## 8. Large classes

**VAR Grade for Windows** can handle classes of about any size. However, there are some things you can do to make using the program easier. The first thing you should do is break the class into sections. Editing a class of 1000 students is easier and sometimes faster if only one or two sections are listed rather than all 1000. When editing, select only a couple of sections at a time. When plotting, however, you want to make sure you've selected all the sections.

### (a) Transferring scores between classes

For classes with many sections, teaching assistants often grade the sections. You have two options: have each teaching assistant keep their section in a different **VGW** class, or have one class that each teaching assistant can access. In the second case, you can use passwords to restrict them to only their section.

#### I. One master class

If you keep all the grades in one class, you can either have the teaching assistants hand their grades to you, and have you enter the data yourself, or let each assistant use a password for their section. To specify section passwords, select "**Option Password**". Click on the "**Sections**" button. For each assistant, enter their password. Also enter a master password for the class. The master password allows access to all sections, while the section passwords allow access to only one section. If no password is entered for a section, that section can only be accessed by the master password. One word of warning about passwords: they are not 100% secure, but only decrease the chances of snooping.

#### II. Separate classes for each section

If each assistant has their own class, you need to make a master class consolidates the grades.

There are two ways to consolidate the grades: export grades from their section as a comma and quote file; or directly export them into your class.

In either case, to import the grades into the master class, you need to make sure that (a) the section numbers are exported along with the grades, and that (b) the task (exam) names that the teaching assistants use are *identical* to the ones you use. So, for example, they must all use "Midterm #1" and not "Exam #1" or "Midterm 1", etc. Another potential problem is that the new scores *replace* the

old ones. In other words, the imported scores will supersede any you have for that task.

Other ways to export and import scores are suggested above.

### **(b) Moving students between sections**

You can move all the students from one section to another very easily. Select “**Edit Move a section**”. You will need to enter the old and new section numbers. Every student in the old section is moved into the new section.

You can move individual students from one section to another when editing by just changing the section number. See Section C4 for more details.

### **(c) Deleting Sections**

You can remove one or more sections of a class. Select “**Edit Remove a section**”. Every student in the section or sections that you choose will be removed from the class. All students in the class can also be deleted from this menu.

## **E. Sorting Students**

Students can be sorted by a variety of methods, including by name, ID, by the scores of any defined task (high to low score), by database items, by no sorts at all, or even randomized. In addition, you can sort using or not using sections. For example, you can sort by name, or by name and section. By default, the program sorts by name and section.

You can change the method of sorting by selecting “**Option Sort order**”. Click the mouse on the desired option. If you select tasks or database items, you need to specify the one to use for the sorting.

The program will continue to use that sorting method until told differently. Sorts are not done until you are finished altering data, so if you add new students, the names will not be alphabetized or sorted until you exit that particular window.

## **F. Transferring Names to a New Class**

If you have entered names into one class, and need to have the same list of students entered into a second class, there are several ways it can be done.

### **1. Export to new classes**

The easiest way to export students is to directly export the data.

- Select “File Export Directly to another class”.
- Select the tasks to export. If the task doesn’t exist in the new class, the new one will be defined like the exported one. If you select no tasks, only the names, IDs, and sections will be exported.
- Select the class to export to. Choose a class, or type in the name of a new one. The data will then be written to the new class. If the class already exists, the old class will have the new students added.

## 2. Saving with a new name

If no task scores have been entered, you can simply save the data of the first class, then save them again with a different name. Saving with a new name is done by selecting “File Save as”.

*WARNING: When you change the name of the class, the program assumes that the new name is the one you are using. So if the first class was English, and the second Physics, the program will use the Physics name as the class currently in memory.*

## 3. Saving with a new name, part 2

If tasks have been entered into your class, you can do the same procedure as the previous section, except that you then need to remove all the tasks from the second (new) class. Save the data to a new class, then select “Task Remove”. Then select all the tasks.

## 4. Export as a comma and quote file or flat file

You can export the class names as either a comma and quote file or as a flat file. The comma and quote files are the best choice. Then you can read the names back in to the new class. This method is the most complicated, but the most flexible, as well.

To export data as a comma and quote file:

- Select “File Export Comma and quote”.
- If database items are defined, select the items to export.

- Select the tasks to export.
- Select how the names and IDs will be exported. If you select “Cancel”, task and database titles will be exported, but not any students.
- Select the order of the items to be exported. At least one item must be selected.
- Last, select the name of the file.

The next step is to import the data into a new class. First, enter the class you want to add the data to. Follow the directions in Section D3 for importing comma and quote files. The data will be imported into the program. Database items will be correctly defined, but any tasks imported this way will end up either number or letter tasks. The other tasks will be converted to one of these types.

For exporting to flat files, then importing into a new class, the procedure is the same, except that you start by selecting “File Export Flat files”, and import as a flat file as described in Section D1.

*Example:* To export students from the English class to the Physics class, do the following:  
When in the English class, save the class as a comma and quote file called physics.txt.  
Next, exit the English class, and choose the Physics class. Then tell the program to import a comma and quote file, and choose the file physics.txt.

## V. DEFINING TASKS

Tasks are graded assignments for students. They can be homework, exams, tests, papers, or anything else that is graded. They are also mathematical manipulations of other tasks. There are fourteen different predefined types of tasks in **VAR Grade for Windows**. The tasks are divided into two categories: (1) Regular or input tasks are assignments: those into which you enter grades. There are three types of regular tasks: letter, number, and extra credit. (2) Special or calculated tasks are mathematical manipulations of other tasks. These include sums, averages, percentages, weights, etc. There are eleven types of special tasks.

Tasks are defined by selecting “Task Define”, then selecting one of the fourteen tasks. The first three listed are regular tasks. The others are special tasks. You cannot define any special tasks until at least one regular task has been defined. The three types of attendance tasks cannot be defined until some attendance days have been defined.

### A. Regular Tasks

You can enter tasks as numbers (Number tasks), you can enter them as letters (Letter tasks), or you can enter them as extra credit (Extra credit tasks). A task needs a name, and, if it is a number or extra credit task, the total possible points for that task. The total is used for calculations, and for checking data is entered.

#### 1. Number tasks

Number tasks are ones in which you enter numbers as the grades. The maximum possible points for any regular task is one million! The program checks the scores of every student when you put in the scores to make sure they do not exceed the maximum possible points. If the score does exceed the total points, that score is rejected, and a new score must be entered. When a score is rejected, a minor beep is emitted by the computer (the beep can be off, if desired). This validation of scores reduces the number of errors made while entering student scores. However, you can also turn off this data validation by selecting “Edit Options”, and unchecking “Verify scores when entered”.

You can also specify the minimum points for tasks. Student scores that are below the minimum points are ignored by Special tasks. For example, if the minimum points is set at 0 (the default), scores below 0 will always be ignored when calculations are done. The minimum points is set by selecting “Task Options”, then entering a new number in “Minimum allowed points”.

##### (a) Defining a number task

- Select “Task Define”, then “Number”.
- A list of tasks that have been already defined are listed. You need to type in a new number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.
- Next, you are asked for the name and total points of the task. You can enter a name of up to 24 characters. If you are replacing an old task, the old name and points will be suggested, otherwise the name is blank and 100 points is suggested for the total.
- If you select “Add more”, that task is defined, and you can define another number task. Otherwise, selecting “Done” defines the task, but exits you back to your starting window.

#### **(b) Changing the total points**

If you decide to increase or decrease the total points of a number task, you can do so without changing the student scores. You need to define the task, as described above in Section a. However, when you reach the total points, enter a new number. Tell the program to replace the old task. You are then asked whether to delete the current scores. Select “No”.

*For example:* If task 5 is worth 50 points, and you want to change this to 100, you define task 5 by:

- ◇ Select “Task Define Number”.
- ◇ Select task 5 by clicking the mouse on it, then clicking the mouse on “Select”. Next, click the mouse on “OK”.
- ◇ You will be asked for the name of the task (with the old name shown), and the total points of 50. Change the 50 to “100”, and select “OK”.
- ◇ Select “Replace” the existing task.
- ◇ Finally, you will be asked if the student scores should be deleted. At this point, select “No”.

The new total points will be 100, and the student scores will be the same.

*WARNING: If you decrease the total points, student scores greater than the new total will be reduced to that new total.*



### (c) Data validation

When task scores are entered, they are checked to see if the score is less than or equal to the total points of the task. If the number you enter is too high, the program makes a small noise. This feature catches many typing errors. The noise only occurs when the score is too high for the task.

However, in some cases, you might want to let a student have more points than the maximum. To do this, you can turn off the data validation. Select “Edit Options”, then uncheck “Verify scores when entered”.

The next item on the Edit Options screen is the option that allows you to turn the beeping off. Turning the option off will prevent the beeps, but not the data validation.

## 2. Letter tasks

Letter tasks are ones where the scores are entered as letters. The program converts them to numbers using the grade values, but the scores can be entered, printed, and manipulated as letter grades.

The standard or default grading system is an A with a grade value of 4.0, a B is 3.0, a C is 2.0, etc. Pluses are 0.3 points higher (except A+), and minuses are 0.3 points below the grade (e.g. B- = 2.7; C+ = 2.3). You can, however, define any grading system that you want, including the names and points for each grade, as long as the total grade types do not exceed 50. In fact, if you want to change the grading system, you should change the grade names and values before you define any letter tasks. See Chapter X.

If you do change the grading system, you should do it *before* you enter any letter grades. The program stores the grades as numbers, not as letters. Thus, changing the names of letter grades after some letter grades have been assigned may result in incorrect scores. You can change the grade values for the letter grades at any time (see Chapter X, Section B).

If you mix letter and number grades, you may want to change the letter grade values. If an A is 4 points, a B is 3 points, etc., an A is 100%, a B is 75%, etc. Some people redefine them as A = 90% (or 95%), B = 80% (or 85%), etc.

### Defining a letter task

- Select “Task Define”, then “Letter”.

- A list of tasks that have been already defined are listed. You need to type in a new number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.
- Next, you are asked for the name of the task. You can enter a name of up to 24 characters. If you are replacing an old task, the old name will be suggested, otherwise the name is blank.
- If you select “Add more”, that task is defined, and you can define another number task. Otherwise, selecting “Done” defines the task, but exits you back to your starting window.

### **3. Extra credit tasks**

Extra credit tasks are number tasks that, when the points are added to a number task, the total points of the number task are not increased. Extra credit tasks are treated in special ways by the program. The extra credit tasks need to have the total points be specified, just like number tasks, to check for incorrect data input. However, this number is used only to check data entry. Special exams, described below, count extra credit tasks as 0 total points, and are counted other special ways by some special tasks.

*WARNING:* In the case of discarding the lowest of a list of tasks, if the extra credit score is one of lowest scores, it may be discarded. This may not be what you want, so you should consider this potential problem before you include extra credit tasks in the definitions of Highest Tasks.

Extra credit tasks are defined just like number tasks, except that you start the process by selecting “Task Define” then “Extra credit”.

### **B. Special Tasks**

The second major type of tasks are called special or calculated tasks. Special tasks are sums, averages, highest, percentages, etc., of other tasks. You do not enter scores into special tasks. Instead, you define the task as the sum (average, percentage, highest, etc.) of other tasks. The program then calculates the scores for you. These scores are recalculated each time you change student scores.

*Note:* Unentered tasks can either be ignored by special exams or they can count as 0 total

points. Which of these the program will use is set by selecting “Task Options”, then checking or unchecking “Unentered scores count as zero”. Scores lower than the fewest allowed points will always be ignored, regardless of how you set this program option.

One limitation for defining special tasks is that they can be defined only as tasks acting on tasks with *lower* numbers. Allowing you to add or sum tasks with the same or higher numbers could result in circular definitions that could confuse the computer and produce nonsense or variable results. A second limitation applies only to sums of tasks. The total points allowed for a “Sum task” (like task #30 above), is 9,999,999 (less than 10 million points).

There are two more options of interest. Scores can be truncated or rounded. A score of 95.5 would be truncated to 95 and rounded to 95.5. Rounding or truncation is set by selecting “Task Options”, then checking or unchecking “Truncate scores”. The second option is the number of decimal places to display for tasks. These are also on the Task Options dialog. The rounding or truncation is done to however many decimals you specify. In other words, if you set the decimals to 1, 95.5 would round to 95.5 and truncate to 95.5. A score of 95.55 would round to 95.6, and truncate to 95.5.

NOTE: Tasks can be removed by selecting “Task Remove”, or by redefining the task. If you select “Task Remove”, choose the task number or numbers from the list of tasks. Since student data for the task(s) is removed, you are asked again to make sure that you really want to remove the selected task(s).

## 1. Sum tasks

These are tasks that are the sums of other tasks.

Example: Task 5 can be the sum of task 1 + task 2 + task 3.

Example: Task 12 is the sum of tasks 3, 5, and 7. Each has 100 possible points. Mary had 90 points on task 3, and 85 on task 5. She has 175 of 200 points. When her score for task 7 is entered, there will then be 300 total points possible for her.

Example: At the end of the term, you probably want to add up all the task scores in the class. You can specify a task, e.g. #30, as the sum of all the previous tasks (#1 through #29). Task #30 is a “Sum task”.

You can, for example, define one Sum task as the sum of all the quizzes, another as the sum of all the tests, and a third as the sum of all the homework. You can define another

task as the sum of the previous three sum tasks to therefore sum all the scores.

NOTE: The sums of all the tasks in a Sum task must be less than 10 million points.

### **Defining Sum tasks**

- Select “Task Define”, then “Sum”.
- A list of tasks that have been already defined are listed. You need to type in a new number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.
- Next, you are asked for the name of the task. You can enter a name of up to 24 characters. If you are replacing an old task, the old name will be suggested, otherwise the name is blank.
- You next select the tasks you want to sum. When you’ve selected all you want, select “OK”.

## **2. Average tasks**

Average tasks are the numerical average of other tasks. For example, Task 5 can be the average of tasks 1 through 4. If you use letter grades, the averages are done using the grade values, and are reported as numbers, not letters. Any tasks that have unentered scores are *NOT* averaged, so the averages reflect only those tasks for which the student has had scores entered.

Example: Task 5 is defined as the average of tasks 1 through 4. Suzy has a score of 90 on task 1, 80 on task 2, and 70 on task 3. Task 4 hasn’t been entered yet. Her average is 80 (not 60, as  $240 / 3 = 80$ ). If she gets a score of 88 on task 4, her new average is 82.

Average tasks are defined the same way as Sum tasks, except that you start by selecting “Task Define”, then “Average”.

## **3. Percentage tasks**

These are tasks that average other tasks, and which then convert those averages to percentages.

*Example:* If task 10 is the percentage of tasks 5 and 6 (which each have 100 points), the program adds up the scores from 5 and 6, then divides the scores by the total points of 5 and 6 (200 points). The score is then converted to a percent by multiplying by 100. Hence, the scores range from 0 to 100 percent.

Like sum and average tasks, any tasks that have not yet been entered are not used in the conversions, so the percentages reported for the students are only for those tasks that have already had scores entered.

Percentage tasks are defined the same way as Sum tasks, except that you start by selecting "Task Define", then "Percentage".

#### 4. Highest tasks: Throwing out the lowest scores

Highest tasks throw out one or more of the lowest scores of a list of tasks.

*Example:* You can throw out the lowest two out of ten tasks. Task 11 can be defined as the sum of the 8 highest scores on tasks 1 through 10.

Highest tasks discard the lowest one or more scores, and can report the number as a sum, average, or percentage. Scores that are unentered are discarded first, then the lowest scores that have been entered are discarded. For weight tasks, scores that have the lowest percentages are discarded first, not those with the lowest points.

*Example:* Suppose you have a weight task worth 65 points, and defined as the weight of tasks 1 through 5, with weights of 10, 10, 15, 20, and 20, respectively. If a student scores a 70% on task 5, and an 80% on the other 4, the score from task 5 will be discarded. The student would end up with 80% of 65 points (52 points). No matter which task the student scored a 70% on, if all the others are 80%, the score would end up as 52 points.

##### Defining a Highest task

- Select “Task Define”, then “Highest”.
- A list of tasks that have been already defined are listed. You need to type in a new number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.
- Next, you are asked for the name of the task, and how many tasks to discard. You can enter a name of up to 24 characters. If you are replacing an old task, the old name will be suggested, otherwise the name is blank. You also have to select how the data will be reported. The numbers can be averaged, summed, or a percentage calculated.
- You next select the tasks you want to use. When you’ve selected all you want, select “OK”.

#### 5. Weight tasks

You can define a task as a weight of other tasks. Each task in a list is given a relative

weight. The relative weights can be percentages or arbitrary numbers. You also specify the total points for the weight. This gives a “true” weighting of tasks with a minimum of effort by you.

*Example:* You can define task 20 as the weight of task 10 with a weight of 0.2, task 11 with a weight of 0.3, and task 12 with a weight of 0.5. This is equivalent to saying that task 10 is 20% of the grade, task 11 is 30%, and task 12 is 50%. In fact, giving weights of 20, 30, and 50 would give the same identical results!

If unused scores are not counted as 0 points, unentered scores are ignored by weight tasks, and the weights adjusted.

*Example:* If you have three exams in your class, each worth 100 points, and want to weight them 30%, 30%, and 40%, you can define task 4 as the weight of task 1 with a final weight of 30, task 2 with a weight of 30, and task 3 with a weight of 40. If no score has been entered for task 3, the grade would be weighted 50% for task 1 and 50% for task 2 (since each is 30 of a total weight of 60).

Now for the math! The calculation for the weights is done by taking the weights you give for each task and dividing them by the total points for each task. These numbers are then added up and adjusted so that the maximum possible points is equal to the total points of the weight task. For the example above, assuming you want the total points to be 100, the students scores are calculated as:

$$\text{score} = 100 * (((\text{Task \#1 score}) * 30 / 100) + ((\text{Task \#2 score}) * 30 / 100) + ((\text{Task \#3 score}) * 40 / 100))$$

The parentheses denote calculations that are kept together. In the example, “\*” means “multiplied by”.

### **(a) Discarding scores**

If you discard scores in a weight task, the other weights are adjusted accordingly. The first tasks discarded are the ones with the lowest percentage scores.

*Example:* If you weight tasks 101 through 106, each with a weight of 10, and discard one task, the other five are each worth 20% of the total score.

### **(b) Defining a weight task**

- Select “Task Define”, then “Weight”.
- A list of tasks that have been already defined are listed. You need to type in a new

number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.

- Next, you are asked for the name of the task, the total points for the task, and how many tasks to discard. You can enter a name of up to 24 characters. If you are replacing an old task, the old name will be suggested, otherwise the name is blank.
- You next select the tasks you want to use. When you've selected all you want, select "OK".
- Last, you need to specify the relative weights for each task. The simplest is to have the numbers add up to 100 or 1, but any numbers can be used. The maximum allowed weight for a task is 1000.

## **6. If task**

If tasks let you require an assignment or type of assignment. The task has a criterion that must be passed for a score to be used. For example, you can require that a student receive 60% or more in order to pass an assignment. Students that don't meet the criteria receive zeros.

To require an assignment, you need to define an If task that checks the required assignment. Then, use the if task to calculate scores (the calculated task must be a Weight, Highest, Average, Sum, or Percentage task). If you then use the calculated task in a Final task and a student doesn't meet the criteria the student grade will be a fail. The required assignment must be one of the grades used in the calculations for the task used to assign final grades. To further explain how this works, examine the examples below.

### **(a) Defining an If task**

- Select "Task Define", then "If".
- A list of tasks that have been already defined are listed. You need to type in a new number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.



- Next, you are asked for the name of the task. You can enter a name of up to 24 characters. If you are replacing an old task, the old name will be suggested, otherwise the name is blank.
- You next the task you want to use for the if criterion.
- You next select the criterion. You can select a value that is greater, greater than or equal, equal, lesser than or equal, or lesser than a value. If the task you selected uses letters, you enter a letter grade for the value, otherwise enter a number.

If the task that is checked is a task that uses letters, the If task will also contain letter grades. Otherwise, the task contains numbers.

If you list scores of an if task, students that meet your criteria will keep the scores of the task that is checked. Otherwise the score will be a 0 (or a failing letter grade).

### **(b) Examples of If task definitions**

Example A: Students in my class have two tests, one term paper, and one creative writing assignment. The student must receive a term paper grade of a C or better to pass the class. The tests each count for 30% of the grade, the term paper 30%, and the other assignment is worth 10%. How do I set up a class to do this?

Answer A: Define four tasks. Task #1 is test one, task #2 is test two, task #3 is the term paper, and task #4 is the writing assignment. Next, define an If task as task #5. Make it check task #3, select a criterion of “>=”, and a letter grade of “C”. Next, define a Weight task as task #7. For the weight task, use tasks #1, #2, #4, and #5 (the If task, not task #3). Assign weights of 30, 30, 10, and 30, respectively. To assign final grades, define task #8 as a Final task that uses task #7 to assign grades. Any student that received a grade below a C on the term paper will be assigned a final grade of an F.

Example B: Sometimes I assign book reports on a pass-fail basis. In other words, in addition to tests, etc., students must do two book reports to pass the course at all. Is there a way to do this?

Answer B: Define the book reports. Make the reports worth one point each. Next, define a Sum task to add up the scores. Next, define an If task, using the Sum task, and requiring “>=” 2 points. Next, define a Weight task, using the book reports as well as any other assignments. When assigning a weight to the book reports, assign it a weight of “0”. Last, define a Final task, using the Weight task. If a student did not turn in two book reports, the student will receive a failing grade.

Example C: If example A, above, a student receives a failing grade if the assignment

does not have a “C” or above. How do you do define the tasks if you want a student to receive a failing grade on the term paper if a “C” isn't earned, but not have the student automatically fail the class?

Answer C: The difference would be to define one extra task (#6), that would be the Sum of task #5 (the If task). Next define the Weight task as above, except use task #6 in the definition instead of task #5. A student will receive a failing grade on task #5 only if the score was lower than a C.

### **(c) If tasks and What-ifs**

If you are calculating what-ifs, having either an If or User task in the definition will force the program to estimate the required grades. The estimates calculated are a maximum of 0.1% above the minimum score required, so if a student requires 70% to achieve a grade, the program may report a needed score as high as 70.1%. Again, if more than one assignment is missing, the scores required are an average of what is needed. It may be possible for a student to score below that average and still make the desired final grade if an If or High task score is part of the grade.

## **7. User task**

This task lets you enter a mathematical formula that is used to calculate student scores. This gives the user task a spreadsheet-like quality. It also lets you define a task that would be difficult or impossible with the other types of tasks.

### **(a) Defining a User task**

- Select “Task Define”, then “User”.
- A list of tasks that have been already defined are listed. You need to type in a new number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.
- Next, you are asked for the name of the task. You can enter a name of up to 24 characters. If you are replacing an old task, the old name will be suggested, otherwise the name is blank.
- You must next define a formula for the definition of the task. Common mathematical symbols are used, as are parentheses. In addition, percentages can be designated. There is a list of tasks, and a calculator-like keypad that you can use for the formula, or you can just type in what you want. Section b has more information on defining the formulas. The basics:
  - \* A number in square brackets is a task number.
  - \* A ‘%’ sign in front of a number in a square bracket means percentage of the task.
  - \* Items in parentheses are calculated first.

- \* Calculations are done from left to right.
- \* Clicking on the buttons on the right put the value or task into the formula at the bottom of the screen.

### (b) Defining formulas

For the following,  $N_1$  and  $N_2$  are any numbers, and T is any task.

<u>Command</u>	<u>Definition</u>
$N_1+N_2$	Add the items $N_1$ and $N_2$
$N_1-N_2$	Subtract two items
$N_1/N_2$	Divide $N_1$ by $N_2$
$N_1*N_2$	Multiply $N_1$ and $N_2$
[T]	Task score, e.g., [1]+[2] adds the scores from tasks 1 and 2
[%T]	Percentage of a task, [%1] is the percentage of task 1.

The order of precedence is parentheses first (from left to right), then the other commands. The numbers are calculated from left to right. You can have any number of parentheses.

#### Examples:

<u>User definition</u>	<u>Result</u>
$2*10-1$	19
$2*(10-1)$	18
$2*[1]+5*[5]$	The score is doubled on task one, then five added, then that total is multiplied by the score on task five.
$(2*[1])+(5*[5])$	Adds two times the score on task one and five times the score on task 5.
$2*[1]+(5*[5])/2$	Double the score on task one, then adds five times the score on task five, then divides that result by two.
$2*[1]+(5*[5]/2)$	Doubles the score on task one, then adds to it five times the score on task five divided by two.
$2*([1]+5)*[5]/2$	Adds five to the score on task one, doubles that number, then multiplies it by the score on task 5, and divides the result by two.
$2*[%1]+(5*[%5])$	Doubles the percentage score on task one, then adds five times the percentage score on task five.

## 8. Attendance task

Attendance tasks automatically calculate a score based upon a student's attendance. You need to designate which attendance days to use, and the values for each type of attendance. You can also discard attendance days, effectively allowing students to be absent a certain number of times.

NOTE: if no attendance has been defined, you cannot define an attendance task, and that option won't be listed on the Define Task Menu.

### Defining an Attendance task

- Select “Task Define”, then “Attendance”.
- A list of tasks that have been already defined are listed. You need to type in a new number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.
- Next, you are asked for the name of the task, the total points for the task, and how many tasks to discard. You can enter a name of up to 24 characters. If you are replacing an old task, the old name will be suggested, otherwise the name is blank.
- You next select the attendance days you want to use. When you've selected all you want, select “OK”.
- Last, you enter the values for each type of attendance, and how many days to discard. Values shown on this menu are the default ones, defined on the Attendance Menu. They can be altered any way you want.

## 9. Attendance extra credit task

Attendance extra credit tasks are defined just like the attendance tasks, but treated like an extra credit task, which counts as 0 total points when used to calculate other special tasks.

## 10. Attendance total task

Attendance total tasks are used to add up different types of attendance. You can tell **VGW** which *types* of attendance to total. Each day of attendance of each selected type is counted as 1 points. They are, therefore, significantly different from the previous

attendance types. These numbers can then be used for calculating grades, or just for printouts.

### **Defining an Attendance total task**

- Select “Task Define”, then “Attendance total”.
- A list of tasks that have been already defined are listed. You need to type in a new number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.
- Next, you are asked for the name of the task. You can enter a name of up to 24 characters. If you are replacing an old task, the old name will be suggested, otherwise the name is blank.
- You next select the attendance types you want to use. When you’ve selected all you want, select “OK”.

## **11. Final task**

Final tasks assign letter grades based upon the scores of another task. These scores are assigned only after cutoffs have been designated. Cutoffs are designated either by defining automatic cutoffs *before* you define the task, or by assigning new cutoffs, after you have defined the task. Automatic cutoffs are defined by selecting “Grade Cutoffs”, then “Assign cutoffs”. Assigning cutoffs after the task is defined is done in the “Grade Assign final grades” menu choice.

The final grades are recalculated each time new scores are entered. If you don't want the grades recalculated, you can either change the task to a letter task after the original assignments, or you can transfer the scores to a new letter task.

### **(a) Final task problems**

If the student grades are all A's, I's, or F's, the cutoffs for the grades were probably not defined before the final task was created. You can either define them for that task only by selecting “Grade Assign final grades”, or you can define automatic cutoffs and then redefine the final task. Note that when you define a final task, it inherits the current cutoffs. Changes to the cutoffs made after you have defined a final task do not affect that task. This allows you to use different cutoffs for each final task.

### **(b) Defining a Final task**

- Select “Task Define”, then “Final”.
- A list of tasks that have been already defined are listed. You need to type in a new number, or select a task that has previously been defined. If another task already has the number that you select, you are asked if you want to replace the old task, or insert the new one. Inserting moves all tasks one number later until an unused number is encountered.
- Next, you are asked for the name of the task. You can enter a name of up to 24 characters. If you are replacing an old task, the old name will be suggested, otherwise the name is blank.
- You next select the task you want to use for assigning grades.

NOTE: If there are other types of special tasks that you would like to have the option of using, send a note to us and explain how this type of task would be calculated. We will try to accommodate you.

### **C. View definitions**

Although the list of tasks that is shown when defining or specifying tasks show some of the significant data for the task, at times you want to get more information. The entire definition of the task can be viewed by selecting “Task View definition”. The procedure is as follows:

- Select “Task View definition”.
- Select the tasks whose definitions you wish to view.

You now view a window with the definitions listed. Besides the name, total points, and the type of task, several other types of data may be shown: cutoffs, weights, lists of tasks used in the definition, formulas, and grade or attendance values.

You can select “Tasks” to view different definitions. If you want to change some of the wording, you can select “Output terms”. To print the window, select “File Print”. This window can be viewed with any Windows font. Changing the font can be done by selecting “File Fonts”.

To exit, select “File exit”.

## D. Manipulating Tasks

Tasks can easily be moved, copied, removed, and so forth. You can also transfer scores to a database item or from a database item to a task score.

### 1. Renaming tasks

If all you want to do is to rename a task, select “Task Alter or move”, then “Rename”. You just select a task, then type in a new name.

### 2. Copying tasks

Tasks may be copied to new tasks. The entire task, including its definition, is copied to another location. In essence, the task is duplicated, including student scores. The method is:

- Select “Task Alter or move”, then “Copy”.
- Select the task to copy.
- Select a new number for the task. If the task number you choose already exists, you will be asked whether to replace the task, or insert the copy before the existing task.
- Last, you are asked for the name of the new task.

*Example:* If the task to be copied is a number task, the new task will have the same total points and student scores as the first task.

*Example:* If the task to be copied is defined as the percentage of tasks 1 through 3, the new task will also be defined as the percentage of tasks 1 through 3.

### 3. Moving tasks

Tasks may be moved to another place. The entire task, including its definition and student scores, is moved to the new location. The method is:

- Select “Task Alter or move”, then “Move”.



- Select the task to move.
- Select a new number for the task. If the task number you choose already exists, you will be asked whether to replace the task, or insert the task before the existing task.

When a task is moved, the task is renumbered, and all the special tasks that use that task are redefined to refer to the new number, as well.

#### **4. Transfer a task definition**

Transferring a task definition transfers the definition of a single task, but *not* any student scores (which will be set as not yet entered). However, if you transfer a special task, the grades are recalculated. If the task definition transferred is a regular task, this option is analogous to copying the task and then clearing the student scores. The method is:

- Select “Task Alter or move”, then “Transfer definition”.
- Select the task definition to copy.
- Select a new number for the task. If the task number you choose already exists, you will be asked whether to replace the task, or insert the task before the existing task.

#### **5. Transferring task scores**

Task scores may be transferred to a new task. When tasks are transferred, the student scores are transferred but the task definition is not. The task where the scores are transferred must be defined as a number or letter task. The student scores are copied from the first task to the second.

- Select “Task Alter or move”, then “Transfer scores”.
- Select the task to copy.
- Select a new number for the scores. The new task must exist, and be a number, letter, or extra credit task.

#### **6. Copy a task score to a database item (and vice versa)**

If you have an item defined as a database item, you can transfer the value to a task score

to that item. If the database item doesn't accept numbers or grade names, the values will be blank (unentered). Otherwise, the database items will contain the numbers or grade names. Transferring to a string database item would allow you to annotate the scores, e.g. "92--great score". The transfer is done as follows:

- Select "Task Alter or move", then "Transfer scores to db item".
- Select the task to copy.
- Select a database item for the scores.

The opposite transfer can also be done: transfer database items to a task. In this case, the task must be a regular task (number, letter, or extra credit). The database item must contain numbers if you are transferring to a number or extra credit task, or else the student scores end up as unentered (blank). If you transfer to letter tasks, the database item must contain the same values as the grade names. Otherwise, the student scores become unentered. This transfer is done as follows:

- Select "DB Transfer db values to task".
- Select a database item to copy.
- Select the task that receives the scores.

## **7. Clearing student scores**

Clearing scores makes the student scores unentered. It can only work on regular tasks (number, letter, extra credit), because the others are calculated.

- Select "Task Clear scores".
- Select the task to clear. Only regular tasks are listed.
- When asked whether to delete the scores, say "Yes".

## **8. Removing tasks**

You can remove one or more tasks from the class. If you have the same students from one grading period to the next, removing all the tasks will start the class over for you, while keeping the students in the class.

- Select “Task Remove”.
- Select the tasks to remove. You can select one or more tasks.
- After you specify the tasks to be removed, the program will ask you if you really want to remove them. If you say “Yes”, the tasks and the student scores will be deleted from the class.

## 9. Change the points of a number task

To change the total points of a number or extra credit task, but nothing else, you need to redefine the task. You will be asked whether to delete the old grades, select “No”. The new total points will be there, along with the old grades.

## 10. Task options

There are several options that affect how tasks work. These are listed on the “Task Options” dialog. Brief descriptions of their significance follow:

- \* *Unentered scores count as zero*: When this option is checked, scores that have no values (unentered or blank ones), count as 0 points. This option is usually left unchecked until the end of the grading period. At that time, checking the value will “zero” all the assignments that haven’t been turned in.
- \* *Truncate scores*: When this option is checked, scores are truncated. When unchecked, scores are rounded. For 85.55, a rounded score will yield 85.6, while a truncated score will yield 85.5. Truncation and rounding are done to the number of decimals specified in “decimals for display” discussed below.
- \* *Minimum allowed points*: Students scores below this value are always counted as unentered. Unless unentered scores are set to zero, unentered scores and those below the minimum are equivalent. Entering students scores below the minimum is the best way to excuse assignments for the students.
- \* *Decimals for display*: This option affects printing and displaying of numbers. It also affects the rounding/truncating option above. Lower numbers allow more data to be printed per page, while lowering the precision in the values, so there is a tradeoff involved.

## E. Examples of Task Definitions

Example #1: I gave three tests. The first was worth 100 points, the second was worth 130 points, and the third was worth 211 points. I want the exams to be worth 25%, 25%, and 50%, respectively, and the total points to be 100.

Solution: Define the tests as Tasks 1 through 3, all number tasks. Define task 4 as a Weight Task. For task 4, make the task worth 100 points. Then pick tasks 1 through 3, and assign task 1 a weight of 25, task 2 a weight of 25, and task 3 one of 50.

Example #2: For each test, I give two versions to my students. Half the students take one version, half the other. I want to keep track of each version separately, yet calculate the final averages for the students in my class.

Solution: For each test, define three tasks. The first two are the two versions of the tests. The third one is an Average Task, which averages the first two tasks. The third task will be the average of one task that has an entered score and one that is unentered, hence equivalent to the entered score.

Example #3: I give a homework assignment, then give students extra credit for turning it in early. At the end of the semester, I discard the lowest homework assignment. However, I don't want the extra credit discarded, just the lowest homework.

Solution: When the homework is graded, define the homework task, the extra credit task, and a third task, the sum of the homework and extra credit. When you discard the lowest homework assignment, use the third task in the definition, not the first two tasks.

Example #4: I want to distinguish between unentered grades and excused grades. How do I do it?

Solution: The program keeps track of the lowest allowed score. When the program is first run, the lowest allowed score is 0. You can, however, enter scores less than 0. You can always enter scores down to -1,000,000. NOTE: You can tell the program the lowest score to use for statistics on the Task Option screen. The program, when calculating statistics, assumes that scores less than the fewest allowed points are unentered scores. The scores you entered, however, are displayed correctly (as scores less than the fewest allowed) when printing or entering scores. Hence, if you enter a -100, it will be ignored for the plots, printing, etc., but written as a -100 when entering data.

Example #5: I want to assign grades to students halfway through the semester. I then want to assign final grades at the end of the semester.

Solution: Define a final task at the half way point that assigns grades based on the first half scores. Then assign grades by selecting "Grade Assign final grades". Define a second final task that assigns grades based on the grades for the entire semester.

Example #6: I defined a number task as having 100 points. I entered grades, then decided to redefine it as having 110 points. Yet, I do not want to lose the previously entered grades.

Solution: When you redefine a task, the program asks whether to delete any previous grades.

If you say no, the grades will not be changed. The only exception is if you decrease the number of points, and a student has more than that number. That student score will be decreased to the new maximum points for the task.

*Example #7:* I give three quizzes each week. Then I define a task as the sum of these three scores. It's tedious to keep defining the tasks each week. Is there an easier way?

*Solution:* Define the tasks for the first week. Thereafter, transfer the task definitions. If the first week tasks were 1 through 4, transfer them to tasks 5 through 8. The new task 5 through 7 definitions will be the same as tasks 1 through 3, and task 8 will sum tasks 5 through 7.

*Example #8:* Halfway through the semester, I like to give the students a handout that tells them their current grades, including a tentative final grade.

*Solution:* Define a task that sums, averages, or calculates the percentages of the scores on the tasks. Define the automatic cutoffs by selecting "Grade Grade cutoffs Assign cutoffs". Use your standard grade cutoffs. Then define a final task, using the previous task. This task will now calculate final grades based upon only those scores that have been entered.

*Example #9:* I set up my class so that tasks 1 through 10 are quizzes, 11 through 20 are tests, and 21 through 40 are homework. I just gave my eleventh quiz. What do I do?

*Solution:* There are two options. When you define the 11th quiz, just tell the program to insert the task before the current task #11. A second option is to move tasks 11 through 40 to higher numbers. For example, move them to tasks 21 through 50.

*Example #10:* I gave three exams, each worth 100 points. However, the third exam should have been worth 150 points. How can I fix this?

*Solution:* Scale the task by selecting "Grade Scale a task", and by telling the program to multiply the scores by 1.5. The new scores will each be 1.5 the original scores, and the exam total points will be 150.

*Example #11:* I used one class for the fall semester ("Fall"), and another for the spring semester ("Spring"). I want to consolidate the grades from the two classes so and calculate a final grade for the year. How can I do this?

*Solution:* One way is to define a third class, "Year". Export the task scores you need for the final grades to another class by selecting "File Export Directly to another class". Export the task scores from each of the two classes "Fall" and "Spring". Make sure that, before you export the tasks to the "Year" class, the names of the tasks from the two classes are different.

*Example #12:* I want attendance to count for 20% of the grade in my class, tests 40%, and homework 40%. In addition, I allow three absences in the semester. How do I do this?

*Solution:* First, define a task that totals the homework (e.g., #20), and one that totals the tests

(#21). Next, define an attendance task (#22). Make the task worth, for example, 100 points. Pick which days to use for the attendance. Then, specify three days of attendance to be discarded. Last, assign weights for each attendance type (e.g., 2.0 for present and excused, 1.0 for late, and 0 for the rest). Next, define tasks that total the homework and tests. Last, define a weight task (#30), and assign the attendance task (#22) a weight of 20, homework (#20) a weight of 40, and tests (#21) a total of 40. To assign final grades, define a final task that uses the scores of the weight task (#30). Whew!

Example #13: I gave one test worth 10 points, another worth 20 points, another worth 100 points, and another worth 500 points. At the end of the semester, I want to count each of them equally. How do I do this?

Solution: Define a weight task, that gives weights for each of the four tasks the same number. For example, weight each of them with a weight of 1.0.

Example #14: I want to mix number and letter grades. However, when I do so, the letter grades don't count right. I want an A to count as 95 points, a B as 85, a C as 75, etc. How do I do this?

Solution: Redefine the grade values by selecting "Grade Grade names and values", then "Change only grade values". When you do this, instead of an A counting as 4.0, make it count as 95. Make a B count as 85, etc. Make the total points count as 100.

Example #15: I want to get an average of four tasks. However, they are worth 10, 20, 50, and 100 points. I want them to count equally in the average.

Solution: If you want the average to be on a 100 point scale, define a percentage task for each, then average them. For example, if the four tasks are numbers 1 through 4, make task #5 the percentage of task #1. Make task #6 the percentage of #2, etc. Then average tasks 5 through 8. It will now be an average. A second method is to define a Weight task, worth 45 points ( $100 + 50 + 20 + 10 = 180$ ; then  $180 / 4 = 45$ ), and weight each of the four tasks equally (a weight of 1 for each).

Example #16: I have tasks that have 10, 20, 50, and 100 points. I want to discard the (relatively) worst score of the four tasks.

Solution: Like the solution to #15, define four tasks as the percentages of the four tasks. For example, define tasks #5 through 8 as the percentage of tasks #1 through 4. Now define a highest task (task #9) that is an average, and discards one of the four tasks.

Note: For examples #15 and #16, you could also define a weight task to do what you want. In the first case, weight each task equally. In the second, weight each task equally, and discard one of them.

Example #17: Sometimes I assign book reports on a pass-fail basis. In other words, in addition to tests, etc., students must do two book reports to pass the course at all. Is there

a way to do this?

Solution: To do this, define each book report as a number task, worth one point each. Next, define a High task that sums all the book reports, and discards two of them. The grades should run from 0 to 2 points. Next, define a User task. Make the task definition as whatever your other grades totals (averages) are, then times it by the (High task score - 11). For example, if you average your grades, and call it task #50, and the High task is task #40, make the User task definition: " $([40]-1)*[50]$ ". Scores less than or equal to 0 are ones that didn't meet the requirement (fails). You may need to set the fewest allowed points to a number less than zero (select "Task Options").

If you don't like negative grades, you can add an extra task instead: Define a Final task (e.g., #41) that uses the High task (#40 in the example). Next, assign grades to the task, using 2 as the top grade, 1 and 0 as failing grades. Then replace  $([40] - 1)$  in the example above with  $([41] / 4)$ . This assumes that your top grade is "4" (an A in the default grades). A second way to do this is to use an If task. Like the first solution, make three book report tasks, worth one point each. Next, define a Sum task that sums the three book report scores. Next, define an If task that passes only scores of two or more (by selecting "greater than or equal" and entering a value of 2). Next, define a special task (e.g., weight), that uses the If task. Last, define a Final task that uses the weight task. A score of 0 in the if task will cause a failure (F). Make sure that unentered scores count as zero, otherwise the If task might have unentered scores for students, which are ignored.

NOTE: Chapter IX, Analyzing the Grades, gives more examples.

## VI. DATABASE ITEMS

Besides names, IDs, section numbers, and tasks, you can enter database items. **VAR Grade for Windows** allows a limited database of up to 200 items, which can be up to a total of 16K in size. These items can include any type of data about a student, including year in school, birthdate, address, sex, age, major, school, etc. There are six different types of items that can be defined. These database items can be altered, printed, sorted, used to filter plots or printouts, and used to define which grading system a student needs.

Database items can be used in all parts of the program. Student names, IDs, and sections are actually special types of database items, and cannot be added or deleted (although the sizes of names and IDs can be changed).

### A. Defining a New Item

There are six types of database items:

- Strings have up to 40 characters.
- Memos are strings of over 40 characters. Note that Memo items are treated a little differently than the other strings due to their length.
- Integers are ordinal numbers from -2,147,483,647 to +2,147,483,647.
- Dates
- Characters are one character long, and must be one of any of 15 possible characters selected by you.
- Booleans are yes or no (true or false).

To define a database item, you select "DB Define". You then select the type you want.

#### 1. String

Strings can be used to enter small amounts of data (since you can enter up to 40 characters). This data can be any ASCII characters. Examples of types that you would want to be string items are addresses, phone numbers, and nicknames. The memory used by each string type takes 1 byte in addition to the number of characters.



### **Defining a string**

- Select “DB Define String”.
- Select a number. This number depicts the position of the item.
- Enter a name. The name can be up to 20 characters long. Also specify the size of the item. The size can be up to 40 characters.

## **2. Memo**

Memos are for entering lots of information about a student. They have between 41 and 254 characters. Because the lengths of the Memo item can be more than the width of the screen, the Memo items must be handled a little differently, for example, printing of the items can take more than one line. Like strings, each Memo item takes up one byte in addition to the number of characters it has.

### **Defining memos**

- Select “DB Define Memo”.
- Select a number. This number depicts the position of the item.
- Enter a name. The name can be up to 20 characters long. Also specify the size of the item. The size can be up to 254 characters, and must be larger than 40.

## **3. Integer**

Integers can be used for numbers above and below zero that do not have decimals. They could be used for grade levels, number of siblings, etc. Integer type items each take up four bytes of memory.

### **Defining integers**

- Select “DB Define Integer”.
- Select a number. This number depicts the position of the item.
- Enter a name. The name can be up to 20 characters long.

## **4. Date**

Dates can be used for date of admission, birthdate, date of leaving, etc. Each date takes up four bytes of memory. When entering dates, you use the format xx/yy/zz, where xx, yy, and zz are the day, month, and year. The date format is on a country by country basis, so xx could be the day, month or year, depending upon your country. The date format used for printing can be changed by selecting “Option Date format”. Dates are printed using this date format. The names of the months and days can also be changed by selecting “Option Date names”.

### **Defining dates**

- Select “DB Define Date”.
- Select a number. This number depicts the position of the item.
- Enter a name. The name can be up to 20 characters long.

## **5. Character**

Characters are used for entering one character. For example, gender has the allowed characters “M” and “F”. The allowed characters are specified by you. The program then checks when you enter data to make sure the correct allowed characters are entered. Character types each take up one byte. If you don’t want the input to be checked, define a string database item of length 1 instead of a character.

### **Defining characters**

- Select “DB Define Character”.
- Select a number. This number depicts the position of the item.
- Enter a name. The name can be up to 20 characters long. Also enter the allowed characters. You can type in up to 15.

## **6. Boolean**

Booleans are used for yes or no data. For example, if they the student met a certain requirement such as an oral report. Boolean types each take up one byte. They are a special case of characters, where the input is set to “Y” and “N”. The allowed input, therefore, are these two characters. Note that you can change the characters used for yes and no by selecting “Option Output terms”, then “Yes and no”.

### **Defining booleans**

- Select “DB Define Boolean”.
- Select a number. This number depicts the position of the item.
- Enter a name. The name can be up to 20 characters long.

## **B. Altering Database Definitions**

The definition of database items can be changed. In addition, you can rearrange them. Just as tasks can be moved and rearranged, so can database items. You can also transfer a task score to or from a database item.

### **1. Rename an item**

To change the name of a database item:

- Select “DB Rename”.
- Select a database item to rename.
- Enter a new name. The name can be up to 20 characters long.

### **2. Alter a definition**

Altering a definition allows you to redefine the item with a different type. If the two data types are compatible, the data is retained. For example, if you defined a date database type, you could redefine it as a string without changing the dates. However, converting a date to a boolean will not succeed: all student data will be blanked. *Note: data that is not valid may be changed or blanked.* As long as these cautions are kept in mind, any type of data can be converted to another!

#### **Altering database items**

- Select “DB Alter”.
- Select a number. This number depicts the position of the item.
- Choose a new type for the data.

- You are then asked for the name and any data needed for the definition.

### **3. Move an item**

You can change the order of database items by moving them. Listing and printing the items is frequently done in numerical order, so changing the order can get your printouts looking better.

#### **Moving database items**

- Select “DB Move”.
- Select the number of the item you wish to move.
- Select the new number for the item.

### **4. Clear student data**

Clearing student data removes all student data from that item. The item is still defined the same way, the student data just disappears.

#### **Clearing database items**

- Select “DB Clear”.
- Select the number of the item you wish to clear.
- Select “Yes” when asked if the data should be removed.

### **5. Delete a database item**

This option removes an item from the database. All student data is deleted for that item, as well as the item definition.

#### **Clearing database items**

- Select “DB Delete”.

- Select the number of the item you wish to clear.
- Select “Yes” when asked if the data should be removed.

## **6. Copy a database item to a task (and vice versa)**

You can transfer database item values to a task. In this case, the task must be a regular task (number, letter, or extra credit). The database item must contain numbers if you are transferring to a number or extra credit task, or else the student scores end up as unentered (blank). If you transfer to letter tasks, the database item must contain the same values as the grade names. Otherwise, the student scores become unentered. This transfer is done as follows:

- Select “DB Transfer db values to task”.
- Select a database item to copy.
- Select the task that receives the scores.

The opposite transfer can be done: transfer the task score to a database item. If the database item doesn’t accept numbers or grade names, the values will be blank (unentered). Otherwise, the database items will contain the numbers or grade names. Transferring to a string database item would allow you to annotate the scores, e.g. “92--great score”. The transfer is done as follows:

- Select “Task Alter or move”, then “Transfer scores to db item”.
- Select the task to copy.
- Select a database item for the scores.

## **C. Database Operations**

Database items can be used in most parts of the program. Below are some examples of this.

### **1. Entering and changing student data**

For the options below, you need to have defined some database items, see Section A.

Student data is entered into a database item one of three ways:

- When entering class data. To list the database items, select the “View” button. A menu will pop down that lists “DB items”. If you select this menu choice, you are asked for the items to view. The database items are listed on the data entry screen after names, IDs, and sections, but before tasks and attendance. When entering the data, each type of database item requires different type of input, so that Integer items accept only numbers, Boolean types accept “Y” or “N”, etc.
- When entering individual data. To list the database items, select “View Database items”. A menu will pop down that lists “DB items”. If you select this menu choice, you are asked for the items to view. The database items are listed on the data entry screen after names, IDs, and sections, but before tasks and attendance. When entering the data, each type of database item requires different type of input, so that Date items accept only numbers, Character types accept your allowed characters, etc.
- Reading in flat or comma and quote files. Data that is listed in flat files or comma and quote files can be imported into **VGW**. See Chapter IV, Section D for more details.

## 2. Sorting

You can sort the class by any database item. The data is sorted according to the type of data, usually either alphabetical or numeric sorts.

- Select “Option Sort order”.
- Select “database item”.
- Select the database item.

## 3. Printing database items

Database items can be printed from all the standard printing menus. The database items are listed before the student grades or attendance.

## 4. Filtering students

You can also filter students through database items before most printing, plots, or statistics screens. You can specify that all students not only meet task score criteria (above/below certain scores), but also they can meet database criteria. For example, if you have a character field for gender (M or F). You can specify that only males, or only

females be plotted. This can be done for any database item.

The “Select Students” option found on many of the window menus, but not the Main Menu, lets you pick students who meet a database criterion. You would select:

- Select “Only students who match values”.
- Select “Match DB items” on the bottom.
- Select a database item.
- Select the db value or values and whether to select students above, below, between, or equal to a value or values.

## **5. Exporting data**

Data that is written to comma and quote or flat files can include the database items you specify. See Chapter IV, Section D for more details.

## VII. ENTERING DATA

Data can be entered into the program in either of two ways. Either the whole class can be listed for a set of data, or data for a single student can be listed. You can switch between these two methods several ways:

- Press <F7> to edit the whole class, <F8> to edit a single student.
- Select “Edit”, then “by Section or class” or “by Individual” on the Main Menu.
- When editing, press the <Right Mouse Button>. If the cursor is on a name, select “Edit individual student” or “Edit by class”. If you are on another type of data, select “Student” first, then one of those two options.

### A. Entering Data

When entering data for the whole class, data is listed spreadsheet style. Each student is listed in a vertical row, while the different types of data are in the columns across the width. The order of the columns is the name, ID, section, database items, and then tasks or attendance. Each of the data types may or may not be listed, depending upon your preference.

When entering data by individual, data for the student is listed vertically. The columns are identical to that of entering class data.

#### 1. Choosing columns

When tasks or attendance dates are defined, they are added to a list of data to show. However, it is usually more efficient to edit a subset of the data. You can specify exactly which data is viewed at any one time by pushing the “View” button on the Class Data Entry screen, or by selecting “View” on the Individual Data Entry screen. In either case, a menu appears that allows you to alter the columns on the screen.

#### Fonts

Selecting the “Fonts” option will allow you to change the font for editing data and printing. You can select any Windows font, as well as its style and size. The font remains in effect until you change it.



## **Name**

Selecting “Name” will turn on or off whether to view names. If the item is checked, names are viewed, otherwise they are not viewed.

## **ID**

Selecting “ID” will turn on or off whether to view IDs. If the item is checked, IDs are viewed, otherwise they are not viewed.

## **Section**

Selecting “Section” will turn on or off whether to view sections. If the item is checked, sections are viewed, otherwise they are not viewed. This option is different than the “Section” button. That button allows you to select which sections to view, while the menu option selects whether to edit the section field.

## **Tasks**

Selecting “Task” lets you choose a different set of tasks to edit. If you have 40 or 50 tasks, using the scroll bars to move to the right task can get tedious. This option lets you choose only some of the tasks to edit.

## **Attendance**

Selecting “Attendance” lets you choose a different set of attendance dates to edit. If you have 40 or 50 dates defined, using the scroll bars to move to the right task can get tedious. This option lets you choose only some of the dates to edit.

## **Minimum widths**

There are two options that are relevant only for class data entry that let you specify that the minimum possible width be used for editing. Normally, **VGW** makes the field width large enough to see both the name of the item and its data. If you specify the minimum width, the field is only wide enough for the data. This will allow more tasks or dates on the screen.

Another option, on the “Edit Options” menu, also can allow more items on the screen: “Max. col. width when editing class”. If you, for example, set the value at 10, the columns will be a maximum of 10 characters wide.

## 2. Moving around the screen

A cell is a datum. It can be a task score, a name, etc. The current cell is highlighted. To move between cells, either the cursor keys, <Tab> and <Shift Tab>, or a mouse can be used.

- The cursor keys are the ones on the keypad, including the arrows and the <Home>, <End>, etc. keys.
- The <Tab> key will move you to the next cell, while the <Shift Tab> moves you to the previous cell.
- The <Enter> or <Rtn> key can do one of three things: move you right, down, or nothing. You can specify which of these three actions to take by selecting “Edit Options”, then clicking on the desired behavior.
- To move with a mouse, just move the cursor to the desired cell, then press the <Left Mouse Button>. You can also use the mouse to move the scroll bars on the right and bottom of the screen.

There are some cells that are protected. This means that you cannot move the cursor to that cell. Calculated tasks (special tasks) are protected because you cannot alter their data. Names, IDs, and section numbers can also be protected. You can protect or unprotect them by selecting “Edit Options”, then “Prevent alteration of ...”.

## 3. Entering data

Cells the you can move the cursor into can be edited. Blank data is data that hasn't been entered. When you type in new data, the previous data will disappear. After entering a cell, if you first type a number or letter, the old data disappears. If you type a special key first (a <Backspace>, cursor, or function key), most or all the data is retained, but the cursor moves. If you want the old data, press <CtrlZ> (undo). The undo must be done before you leave that cell.

The data you can enter is affected by the data type. Below is a list of the data types and the allowed data.

### I. Names

Names can contain any type of character, and can be up to 30 characters long. The maximum length that a name can have is specified in the “Option ID and name

sizes” dialog. If you want to enter an unusual character (e.g., above the ASCII value of 127), simultaneously press <Alt><Shift> and type in the ASCII value with the keypad numbers (this is a Windows method, not specific to **VGW**).

## **II. IDs**

IDs can contain any type of character, and can be up to 30 characters long. The maximum length that an ID can have is specified in the “Option ID and name sizes” dialog. If you want to enter an unusual character (e.g., above the ASCII value of 127), simultaneously press <Alt><Shift> and type in the ASCII value with the keypad numbers (this is a Windows method, not specific to **VGW**).

You can also specify how many digits of the ID to print. This helps mask the identity of students. For example, you could print only the last 5 digits of the ID. This is set on the same dialog as the ID size.

## **III. Sections**

These are numbers, and can be any number from 1 to 65530. No input other than numbers is accepted by the program. Note that you cannot enter section numbers unless “Multiple sections” is checked and you have selected the “Section” menu item from the “View” button.

## **IV. Number tasks**

These are numbers, but can include decimals. The number entered must be less than or equal to the total points of the task unless you have turned data validation off (see below). Numbers that are too high are normally rejected with a beeping sound. You can turn off the sound, or make it louder, by selecting “Edit Options”, then unchecking “Make sound on data input errors”. You can allow more points than the possible points by turning off error checking on that same menu (“Verify scores when entered”).

The minimum allowed points are -1,000,000, while the maximum for any task is 1,000,000. To enter a zero, you must type a “0”.

Cells for tasks that are left blank revert to unentered scores.

*NOTE:* Scores will be listed be the number of decimal places that you designate. The default is one decimal place, but you can change it to 0 (if you don't use decimals) or up to 5 decimal places. The minimum allowed points and the decimals listed can be changed by selecting “Task Options”.

## Inline calculator

**VGW** sports an inline calculator. If you have used computer calculators, you have seen them pop up and look like a hand-held calculator. This isn't necessary. **VGW** does the math calculations without popping up. It doesn't look as fancy, but requires less work to use.

The main purpose of the inline calculator, which is available whenever floating point (decimal) numbers are entered, is to let you add up point totals for a student's assignment. One way to do this is to define a different task for each question or part of the assignment, enter the numbers, then calculate a sum. The inline calculator is much faster and easier, but doesn't save the numbers from each question.

It is called an inline rather than a popup calculator because you only need to press a "+", "-", "/", or "\*" to activate it. The "\*" means multiplication.

If you are entering scores, and haven't added up the points on the exams, the inline calculator is ready to be used. Type in the first score, then a "+". The first score will be moved left, and a "+" sign is shown. You then type in the next score. The score will be automatically calculated. You can then enter another "+", and another score, etc. When you are finished, you can end with a "=", or by using any cursor key to move to another cell. The finished score will be automatically pasted into the original cell.

*Example:* You have a student test. The student received a 10 on the first question, 23 on the second, and 13 on the third. To calculate the score, type in a "10", then a "+". The inline calculator is now at work. Then type in a "23", and another "+". The calculator shows a "33 + ". Next, type in the "13" and an "=". The calculator will disappear, and "46" will appear in the cell. You could also have ended the calculation by typing in a <Down Arrow> instead of an "=".

*NOTE:* The calculator keeps track of decimal places. It will paste in a number that has at least as many decimal places as the numbers you use. If you divide a number, it will retain a minimum of two decimal places.

## V. Letter tasks

If you are entering letter grades, only letters that are in the names of the grades are allowed (e.g., for grades from A to F, only the letters A, B, C, D, and F are allowed).

A blank cell is an unentered score. The grading system can be changed on the “Grade names and values” menu. Up to three characters can be used to specify a letter grade.

#### **VI. Attendance**

If you are entering attendance, only letters that are specified on the “Attendance Options” dialog are allowed (e.g., “P” for present, “E” for excused, “L” for late, etc.). If you want to change the acceptable keys, go to that dialog. Up to two characters can be used to specify an attendance type. A blank value is one of the attendance types (the default is “present”).

#### **VII. Strings and Memos (db)**

String and memos can contain any type of character, and can be up to the length specified when the database item was defined (as long as 254 characters for a Memo). If you want to enter an unusual character (e.g., above the ASCII value of 127), simultaneously press <Alt><Shift> and type in the ASCII value with the keypad numbers (this is a Windows method, not specific to VGW).

#### **VIII. Integers (db)**

Integers are numbers, and can be any number from -2,147,483,647 to +2,147,483,647, and are entered without commas as separators (only numbers and a negative sign are acceptable). No input other than numbers is accepted by the program.

#### **IX. Dates (db)**

Dates are entered as pairs of numbers, separated by “/”, in the format xx/yy/zz, where xx, yy, and zz can be days, months, or years, depending on the format of your country. The date format can be changed by selecting “Option Date format”, which will affect some printouts. However, only “short dates” (with numbers) are used for entering data.

#### **X. Characters (db)**

Characters are fields that accept only one character. The character can be any one specified when you defined the database item. Other characters are rejected. Lower case characters are converted to uppercase ones, so “e” and “E” are considered the same character.

#### **XI. Boolean (db)**

Booleans accept only yes and no input. In the editing screens, booleans are entered by using “Y” and “N” (in English). You can change these values by selecting “Option Output terms”, and then “Yes and no”.

#### **4. Recalculations**

Automatic recalculations are done by the program unless you turn them off. These recalculations are done each time you leave a task cell, or leave either of the two editing windows. Recalculations can significantly slow down the program. To turn them off, select “Edit Options”, then “Automatically recalculate grades”. If you turn off recalculations, they will only be done when you leave a screen, but not when moving between cells.

#### **5. Listing sections**

If you have more than one section in the class, you can change which sections are listed on the Class Data Entry screen by selecting the “Section” button. Select a new list of sections by clicking on one student in each section you want to view.

#### **6. Quick filling in of scores or database items**

On some assignments, most of the students receive the same grade, or need the same value on database items. You can fill in the needed values very quickly if you are on the Class Data Entry screen by moving to the desired column and pressing the “Fill column” button.

All blank items in the column will receive that value. Students that have already had values entered are left unchanged.

#### **7. Locating a value or student**

If you have lots of students in a class, you may have trouble locating a student or value of a cell. There are two ways to locate an item on the current column, and a third one that works just for students:

- Press <CtrlF> (for find). This will find an item on the current column.
- Select the “Locate” button on the Class Data Entry screen, or “Edit Locate Value” on the Individual Data Entry screen. These will find an item on the current column.

- Select the “Find row” button on the Class Data Entry screen, or “Edit Locate student” on the Individual Data Entry screen. This will let you find a particular student. A list of the current students will popup. You select the one you want from the list box.
- Press <CtrlS> (for student). This will popup the same menu as the previous method.

The first two options are different ways to popup the same dialog. The program asks for the value to locate as well as the direction. You can search from the top of the list, from the current position, or backwards starting at the previous position. The search only is done for the column where the cursor is located. If that student or value is not what you wanted, the cursor isn't moved. You don't need to enter the entire value, just one that is unique to what you want to find. For example, entering “Fred” when on the names column will find any student with Fred in the name, including Frederick and Freddy.

## 8. Adding and deleting students

Although there are other ways to add or delete students, you can do this from the data entry windows, too.

- In the Class Data Entry, to add a student you press “Add row” or “Ins row”. Add row adds a row for a new student at the end of the list of students, while Ins row add the new row at the cursor position. You then have to enter a name for the student.
- In the Individual Data Entry, to add a student you select “Insert”.
- To delete students, press “Del row” in the Class Data Entry window, or select “Delete” in the Individual Data Entry.

Both adding and deleting students cannot be done from the data entry windows if the student names are protected.

## 9. Printing

Pressing “Print” will print the current data in the Class Data Entry window. Selecting “File Print” will do the same in the Individual Data Entry window. For either, all the columns and rows are printed.

## 10. Options

You have a large amount of control over how the program works. Selecting “Edit Option” and “Task Option” on the Main Menu will show you most of the options you can change. A quick synopsis of them is listed below. Most of these are discussed in other parts of the manual.

- Alteration of fields: You can prevent names, IDs, and sections from being altered when entering data.
- Column width: You can specify the maximum width of the columns. You can also tell the program to use the minimum possible width. This is the width needed for the data.
- <Rtn> movement: Normally, a <Rtn> moves you to the next cell to the right. You can make a <Rtn> move you to the next cell down, or not move at all.
- Verify entered values: normally, the program verifies typed in number grades for valid scores. You can turn this off, allowing scores above 100%.
- Noise when errors: If values are too large, the program beeps at you. You can turn this off.
- Colors: You can specify the colors to be used by **VGW**.
- Decimal places: you can specify how many decimal places to use in printouts and calculations.
- Unentered scores set to zero: Used in recalculating grades. Unentered scores can be ignored or set to 0 points.
- Truncate scores: Scores can be truncated or rounded. If truncated, 89.96 will truncate to 89.9, while it will be rounded to 90.0 (if rounding to 1 decimal place).
- Minimum allowed points: Normally scores below zero are always ignored by the program. You can set this value to a negative number, if needed. If you using T-scores or Z-scores, you should set the minimum to a negative number like -10.

## 11. Right mouse button

The <Right mouse button> can be used to pop up a menu of things you can do. The



main purpose is to give you a fast and easy way to view and analyze data. It is highly recommended that you get used to using the right button, as it can speed up some of the things you want to do in **VAR Grade for Windows**.

The menu that appears when the button is pressed will change according to the type of data that is being edited. When the items are selected, the current list of columns and students is used to view or analyze the data. When on the data or view screen, you can then alter the data that is viewed.

## 12. Moving around Individual Data Entry windows

There are several quick keys that can be used in Individual Data Entry screens:

- ◆ <AltN>        Next student
- ◆ <AltP>        Previous student
- ◆ <AltB>        Beginning student
- ◆ <AltL>        Last student

These can also be selected on the Individual “Edit” menu.

## 13. Standard Windows editing keys

We must also mention that the four standard editing keys for Windows are used in all parts of **VGW**.

- ◆                    <CtrlZ>        Undo. This undoes the last edit when in that cell or field.
- ◆                    <CtrlC>        Copy. This copies the selected data to the Windows clipboard.
- ◆                    <CtrlX>        Cut. The copies the selected data to the Windows clipboard, then deletes the data.
- ◆                    <CtrlV>        Paste. This pastes data from the Windows clipboard into a cell or field.

## B. Saving Entered Data to Disk

It is strongly recommended that you save the data you have just entered as soon as possible after entering it. Although the program won't let you exit without saving changes, in case of power failures or computer failure, you will have a copy on disk of your new data. The simplest way to save data is to press <F2>. This works the same as if you select “File Save” on the Main Menu.

If you want to change the filename, select “File Save as”, and enter a new name. After you change the name, further changes are made with the *new* name.

NOTE: The file that contains your student data is normally made a “read-only” file. This means that you cannot delete it with the DOS del or erase commands or with standard Windows file managers. You can tell the program to write your files “read-only”, as “hidden” (the DOS dir command will not even list the file), or “normal”. Normal files can be deleted in DOS. The program also makes backup files “normal”, hence they can easily be deleted. Details on how to change the way the program saves the class data are in Chapter XII, Section A.

## 1. Backing up data

It is also recommended that you keep two copies of your class data on disk. To encourage this practice, the “File Copy” option on the Main Menu has been included. We also strongly encourage you to print out your data after entering it.

## 2. Previous versions of VAR Grade

Many of you use, or have used, **VAR Grade**. In addition, you might want to use **VAR Grade** on one computer, and **VAR Grade for Windows** on another. **VGW** will read data from any version of **VAR Grade**. In addition, you can use **VGW** to write version 5 or 6 **VAR Grade** classes. However, if you do write to older versions of the program, some of the newer features will get converted to data that the older version can understand.

WARNING: If you use this option, some of the information in your class may be lost. Specifically, any task or database types that do not exist in the previous versions will be converted to ones that are understandable to that version, which may not be what you want. If you use this option, use only features in this version that exist in the previous version.

## VIII. ATTENDANCE

Attendance dates can be defined and attendance entered for up to 240 different days. There are 13 different attendance categories, with up to 2 characters used to specify each attendance category. Attendance can be used in grades by using one or more of the three types of Attendance tasks (see Chapter V). Attendance is entered in a manner analogous to entering data for tasks. You can also prepare up to two different seating charts for each section of a class. Attendance and seating are found on the Attendance option of the Main Menu.

For style of the dates is affected by how you want the date to appear. For example, you can have the date include the day of week, month names, or just numbers. Select “Option Date format” to change the style. Any attendance dates defined after you change the style will have the new style. However, any dates defined before you change the style will keep the old names.

### A. Date formats and names

**VAR Grade for Windows** can format dates just about any way you could want. Besides many predefined ways, you can enter your own. In addition, you can change the names of the days of the week and the names of the months.

#### 1. Date formats

**VGW** gets the date format from Windows. If you wish to define a new format for dates:

- Select “Option Date formats”.
- There are a dozen different predefined formats. Not all may look different to you, as the differences may only appear under certain circumstances. The major difference between pairs of dates that look the same, may be how days below 10 are displayed. For example, 8/8/94 versus 8/08/94. The first of the two identical items would display the “08”, while the second would display “8” for the days.
- If you select “Your own date format”, you will move to a new dialog. Below we explain how to enter your own system.

#### Defining your own date format

Most of the explanations of how to define a date format are listed on the “Define the Date Format” dialog. The general method is to enter values into the input box titled “New Format”. In general, capital letters denote “drop the leading zero” (“8”), while

lower case letters specify keeping the leading zero ("08").

	<b>Data output</b>			
<b>Type</b>	<b>01-99</b>	<b>1-99</b>	<b>0001-9999</b>	<b>1-9999</b>
Months	mm		MM	
Days	dd	DD		
Year	yy	YY	yyyy	YYYY
<b>Type</b>	<b>Sun</b>	<b>SUN</b>	<b>Sunday</b>	<b>SUNDAY</b>
Days of week	www	WWW	wwwwww	WWWWWWW

Other characters are allowed, as well. The following are examples of formatting dates:

<b>String</b>	<b>Output</b>
wwwwww, MM nnn YY	Monday, 8 Aug 94
YYYY nnnnnnnn mm	1994 August 08
MM NNN wwwwww	8 Aug Monday

## 2. Date names

The names of the days of the week and the months can be altered as follows.

- Select “Option Date names”.
- Select one of the standard languages, or select “Other language” to type in your own.
- If you want to type in your own names, just move to the name you want to change and type in the new name. End with “OK”.

## B. Defining Attendance Dates

### 1. Inserting one date

If you just have one or two dates to define, you can do it easily as follows:

- Select “Attendance Insert one date”.
- Select a number for the date.
- You next see a calendar. Clicking the mouse on a date sets the date in the edit

field seen below the calendar. Clicking on “Today’s date” will set the date as today. You can also click on the edit field and type in whatever you want. When finished, press “OK”.

## **2. Defining many dates**

The previous method would be tedious if you want to define all your attendance dates at once. To define lots of dates, do the following:

- Select “Attendance Define dates”.
- Select the days of the week that the class is held.
- You next see two calendars. The left one is the starting date, the right one is the ending date. Clicking the mouse on a date sets the date in the edit field seen below the calendar. Clicking on “Start today” will set the first date as today, while selecting “End today” will set the last date as today. When finished, press “OK”. Each day between and including the starting and ending dates that is on the selected days of the week will be added to the list of dates.
- Since schools have holidays on different days, and this program is used throughout the world, **VGW** cannot know which days are holidays. You therefore want to remove the holidays and other days the class will not be held. See section 3 below on how to remove dates.

## **3. Removing dates**

If you have defined dates that aren’t needed, you can remove them as follows:

- Select “Attendance Remove”.
- Select the dates to remove.
- Select “OK”.

## **C. Entering Class Attendance**

There are three places where you can enter attendance: on the Class Data Entry window, on the Individual Data Entry window, and from Seating charts. The first two are described in this section, while the third is described in the Seating charts section (D, below).

Chapter VII describes data entry in detail. To enter attendance from either the Class Data Entry or Individual Data Entry windows, you need to select “View”, then “Attendance”. Next, select the desired dates to view.

You can enter the key codes for any of the attendance categories. These categories and the key codes can be viewed and altered by selecting “Attendance Options”, and is described below in Section E.

## **D. Seating Charts**

You can define two seating charts for each section of students. You use a mouse to move the desks around the room. The seating chart printouts will show the seats of both the teacher and the students. You can also enter student attendance directly from a seating chart.

### **1. Defining seating charts**

Defining a seating chart is simple. You select “Attendance Seating Charts”, then “View”. Seating chart 1 will be shown. If you use multiple sections of students, the first section will be shown.

To define Seating chart 2, select “Chart 2nd chart”. To view the first chart again, select “Chart 1st chart”.

To define a chart for a different section, select “Select Section”.

### **2. Moving the desks**

After the chart has been defined, the first thing you want to do is to move the students. The desks can be moved by moving the mouse cursor to a desk, pressing the left mouse button, and, while keeping it down, moving the desk to the desired location. Both the original desk and the moving one change appearance when you move a desk. Release the mouse button when the desk position is correct. There are three menu options that let you easily change student desks:

- **Alphabetize:** This option will place the students in the existing desks in alphabetical order. The order is left to right and bottom to top.
- **Reverse alphabetize:** This alphabetizes the names, but in reverse order (right to left and top to bottom).
- **Randomize:** This moves the students randomly between the desks.

### **3. Seating chart appearance**

You can make changes to how the seating chart looks. These allow enormous customization of the program.

#### **(a) Names and IDs on desks**

You can change how the names or IDs are printed on the desks by selecting “View”, then one of the options.

#### **(b) Shadows and lines**

The desks can have boxes and shadows around them. The default is to have boxes but not shadows. Select “Select” and “Lines on chart” or “Shadows on chart” to change the desk appearance.

#### **(c) Labels**

There are two options that affect the labels on the chart.

- Chart labels: the chart title and the instructor can be changed for the chart by selecting “Select Chart labels”. The default chart title is the class name or section name. The default instructor is “Instructor”. These values are valid only for this chart.
- Output terms: you can change the term “Instructor”, or many other terms, by selecting “Output terms”.

### **4. Printing seating charts**

The seating chart can be printed by selecting “Chart Print chart”. You can also change the font for the chart by selecting “Chart Font for chart” or the font size by selecting “Chart Font size”.

### **5. Copying the chart**

Selecting “Chart Copy” will copy the chart to the Windows clipboard. You can then paste the chart into any other Windows program.



## **6. Entering Attendance**

If you select “Attendance” when editing seating charts, you can enter attendance with the seating chart. You select the attendance day to edit, and then see a window with the attendance. You cannot move the desks around on this screen, but can easily add the attendance. There are several menu items to help you edit.

### **(a) Defining dates**

You can define dates from this screen by selecting “Define date”. You then define an attendance date exactly like the “Attendance Insert one date” option on the Main Menu. See Section A1.

### **(b) Selecting a new date**

Selecting “Attendance” again lets you select a new date for data entry.

### **(c) Viewing names and Ids**

You can select how the names are listed for entering attendance by selecting “View”, then one of the choices.

### **(d) Other charts**

You can select chart 2 by selecting “Chart 2nd Chart”, or chart 1 by selecting “Chart 1st chart”.

You can select a new section for data entry by selecting “Section”, then the desired section.

## **E. Attendance options**

By default, when attendance days are defined, the student attendance is made “Present”. There are thirteen categories of attendance: Present, Excused, Unexcused, Late, Not Entered, Other, Present 2, Excused 2, Unexcused 2, Late 2, Other 2, Activity 1, and Activity 2. These use the characters “ ”, “E”, “U”, “L”, “\_”, “O”, “P”, “C”, “N”, “T”, “Y”, “A”, and “B”, respectively.

The default student attendance and the characters associated with them can be altered on by

selecting “Attendance Options”. In addition, the names of the attendance categories can be changed, as well. The characters associated with the attendance categories are the characters you enter for attendance, and the ones that are printed when printing attendance by section or the whole class.

To change the letters for the different types of attendance, move to the type and pick the new key or keys. You can use up to two keys for an attendance day. To simplify matters, lowercase letters are always converted to uppercase letters. The allowed keys are all the letters, except the lowercase letters, as they are automatically converted to uppercase letters.

One aspect of attendance should be mentioned. The program can add up attendance of various types. One method of adding them up is total days present and total days absent. For these calculations, present includes the attendance types of present, present 2, late, late 2, activity 1, and activity 2. Absent includes excused, excused 2, unexcused, unexcused 2, other, and other 2. Not entered isn't included in either category. If you change the attendance names, try to keep in mind how they are treated with respect to being present or absent.

## **F. Attendance Task Values**

When you define an attendance or attendance extra credit task, the program will suggest value for the number of days to discard and the weights for the various types of attendance. Although you can change these when you define the attendance tasks, you can change the default values by selecting “Attendance Values”. The number of days you can discard is one less than the number of attendance days used to define the task. The weights that you can use may be any number greater than or equal to zero.

## **G. Exporting attendance**

If you have several classes, each of which meets on the same days, you can define attendance dates for one class, then export them to the other classes. The simple procedure is:

- Select “Attendance Export dates”.
- Select the dates to export.
- Select the class that receives the dates.

## IX. ANALYZING THE GRADES

Grades can be plotted and analyzed statistically a great variety of ways. In addition, you can adjust grades a number of ways. The plotting and statistics should help you decide how and what adjustments should be made. In addition, plotting and statistics can inform you about how your class is doing, and how you are doing as well. In other words, you may be able to improve your teaching if you spend the time to understand how your students are doing on various tasks. There are four major ways to plot in **VGW**:

- Histograms: these include bar charts and pie charts. They plot the scores of the class against how many students were in each subrange. These are the most common plots.
- Correlations: these plot the scores of one task against a second one. These are used to check how similar or different students did on the two tasks.
- Individual progress: this plots the scores of a student on a number of tasks. The scores can be plotted with a variety of statistical measures. This shows whether the score is going up, down, or otherwise.
- Class vs. individual: this plots the score of a student versus the scores of the class, and give a good feel for how a student did on a particular task.

### A. Histograms and pie charts

Histograms and pie charts separate the students into groups, then plot the groups. The groups can be particular grades or subranges of scores. In histograms, the number of students in each group are plotted on the Y-axis, and the grades or ranges of scores on the X-axis.

The plot separates the scores into up to 20 categories of scores (10 for pie charts). It also can show the mean, median, and standard deviation of the scores for the task. The numbers of students in each of the score intervals is plotted. The statistics calculated and printed are for those students scores in the chosen range of scores. If the high or low range of scores are changed, these statistical measures will, therefore, also change.

For histograms, the mean, median, and mode are shown on the plot with arrows. The description is above or to the right of the arrow. At one standard deviation above and below the mean, there are additional arrows (listed as "+1", "-1"). If the standard deviations fall outside the range of scores plotted, they will not be displayed.

To plot either of these types:

- Select “Plot Histogram”.
- Choose a task.
- The scores are now plotted. You can change how the scores are plotted by selecting options from the menu.

The following sections describe the changes you can make to get the plot to look how you want.

## 1. Score ranges

The initial plot selects the lowest and highest scores in the class, separates the class into 20 categories, and then plots them. If the task uses letter grades, the plot will separate the scores into each grade. You can select your own high and low scores by selecting “Select Range”, then entering the new values.

A related matter is whether to plot using your predefined grade cutoffs, or to plot the full range of scores. The default, mentioned above, separates the students into 20 categories. The cutoff option separates the students into grades, as defined by your cutoffs, and plots the grades. To change the option, select “Chart Options”. If you haven’t entered grade cutoffs, the full range will be plotted.

## 2. Plotting different students

In addition to score ranges, there are three additional options that affect which students are plotted. The options listed in parts a and b make a new list of students to plot, while ignoring students just adds or deletes students from the current list of students.

### (a) Changing sections

This option is only available if there is more than one section of students in the class.

- “Select Sections”.
- Pick the section or sections that you want to use.

### (b) Selecting students

The options listed here allow you to select any combination of students you wish to

plot.

- “Select Different students”.
- There are five different options available. You can select no students (not a good idea when plotting), select all the students, select students in sections (the same as in part a, select individual students, and select students who match a value. These options are explained in more detail in Chapter IX, Section H.

### **(c) Ignoring students**

Ignoring students prevents them from being used for plots and statistical calculations. Otherwise they are treated like normal students.

- “Select Ignore students”.
- Choose the students to ignore. The students listed are from the current list you have chosen (section, matching values, etc.), so you can refine your choices.

## **3. Selecting a different plot style**

There are six general styles of plots. Four are histograms, while two are pie charts. The style can be changed by selecting “Chart”, and then one of the six styles.

### **(a) 3D Bars**

This is a standard histogram.

### **(b) 2D Bars**

This type is similar to the 3D bars, but the bars look two dimensional instead of three dimensional. It is a histogram.

### **(c) Line charts**

Instead of bars, there are lines plotted. It is a histogram.

### **(d) Tape graphs**

Instead of bars, there are ribbons (or tapes) plotted. It is a histogram.

#### **(e) 3D Pie charts**

If there are too many categories, the pie labels can overlap. We therefore recommend that it be used primarily for plotting letter grades. In general, the only difference between the pie charts and the plots is how the data is presented.

#### **(f) 2D Pie charts**

This is similar to 3D Pie charts, but without the 3D effect.

### **4. Changing the appearance of the chart**

You can change the chart labels, can add lines on a histogram for visually lining up columns, and can choose to print or not print statistics on the chart. In addition, the mouse buttons will also pop up information.

#### **(a) Statistics**

The statistics values are the mean, median, and standard deviation arrows on the histograms. To prevent statistics values from being printed, select “Select Show statistics”.

#### **(b) Lines on chart**

To show lines on a chart, select “Select Lines on chart”. On histograms, horizontal and vertical lines are shown on the chart, while, for pie charts, lines go from the slice to the grade name or range.

#### **(c) Labels**

The chart title, X-axis title, and Y-axis title can be changed by selecting “Select Chart labels”. The default X-axis title is the task name, while the chart title is the class name. The Y-axis title is “Number of students” if a histogram is selected, but nothing for a pie chart.

Selecting “Output terms” allows changing the statistics names. The output terms affect all parts of the program, while the chart labels affect only this chart. If you change to a different task, the labels will change.

#### **(d) Mouse buttons**

The left and right mouse buttons also pop up information.

- Pressing the <Left Mouse Button> while on a bar will show the percentage of students represented in the bar. Pressing the button while on other parts of the chart will show the percentages for all the categories.
- Pressing the <Right Mouse Button> will pop up the statistics values in numerical form. This includes the mean, median, standard deviation, and the high and low scores.

If you print the chart with this data showing on the screen, the data will be printed, as well.

## **5. Printing and copying**

You can print the chart and the data for the chart, and can also copy a chart to the Windows clipboard.

### **(a) Printing a chart**

After you have changed the chart to get it to look how you want, you can print the chart by selecting “Chart Print chart”.

You can select a different chart font. There are five general styles (Roman, Swiss, Modern, Script, and Decorated). Note that the chart fonts are different than the fonts used for printing other data.

### **(b) Printing data for the chart**

The data used to determine the chart can be printed. **VGW** will list the scores in two columns: (a) by scores on that task, from high to low score and (b) class order (dependent upon your method of sorting students). The rank order of the students is also printed, if that option is selected. In addition, the mean, median, and standard deviation are also printed. Select “Chart Print data” to print the data.

The font for the data is the same font used for printing other data. Note that data fonts are different than chart fonts.

### **(c) Copying a chart**

If you select “Chart Copy”, the chart will be copied to the clipboard. You can then paste a chart into any Windows word processor, paint program, and other programs.

## **6. Choosing a new task**

Selecting “Task” while showing a pie chart or histogram will let you choose a new task to view.

## **B. Correlating Different Tasks**

You can plot the scores for any of the tasks against the scores for any of the other tasks. This helps you determine if there were any problems on one or both of those tasks. If the tasks are measuring different things, the scores will be widely scattered. This may be an indication that one or both of the tasks did not measure what you hoped it would. If the tasks are



measuring similar abilities, the scores will tend to cluster near the diagonal. Used correctly, this can be a powerful tool to analyze your grading methods. To do a correlation plot:

- Select “Plot Correlate two tasks”.
- Choose the X-axis task, then the Y-axis task.
- The scores are now plotted. You can change how the scores are plotted by selecting options from the menu.

The correlation coefficient ranges from 1.0 (the scores on the X-axis predict exactly the scores on the Y-axis) to 0.0 (the scores on the two axes are completely random with respect to each other) to -1.0 (the low scores on the X-axis predict high scores on the Y-axis, and vice versa). The closer the coefficient is to 1.0, the more closely the two tasks measured the same parameters or abilities (and hence the students scored more nearly the same on both the tasks). If the correlation coefficient is close to 0 or below it, then students who did well on one task did poorly on the other. This is probably not desirable, and most likely reflects a flaw on one or both tasks.

## **1. Score ranges**

The initial plot selects the lowest and highest scores in the class for each task, separates the class from the high to low scores for both the tasks, and then plots them.

You can select your own high and low scores by selecting “Select X-axis range” and “Select Y-axis range”, then entering the new values.

## **2. Plotting different students**

In addition to score ranges, there are three additional options that affect which students are plotted. The options listed in parts a and b make a new list of students to plot, while ignoring students just adds or deletes students from the current list of students.

### **(a) Changing sections**

This option is only available if there is more than one section of students in the class.

- “Select Sections”.
- Pick the section or sections that you want to use.

### **(b) Selecting students**

The options listed here allow you to select any combination of students you wish to plot.

- “Select Different students”.
- There are five different options available. You can select no students (not a good idea when plotting), select all the students, select students in sections (the same as in part a, select individual students, and select students who match a value. These options are explained in more detail in Chapter IX, Section H.

### **(c) Ignoring students**

Ignoring students prevents them from being used for plots and statistical calculations. Otherwise they are treated like normal students.

- “Select Ignore students”.
- Choose the students to ignore. The students listed are from the current list you have chosen (section, matching values, etc.), so you can refine your choices.

## **4. Changing the appearance of the chart**

You can change the chart labels, can remove the horizontal and vertical lines on the chart, and can choose to print or not print a linear regression line on the chart. In addition, the mouse buttons will also pop up information.

### **(a) Statistics**

The linear regression line is a representation of the correlation of the two tasks. The line, ideally, goes from the bottom left to the top right of the chart, at about a 45 angle. The further the line differs from the ideal, the less closely the two tasks measured the same thing. A horizontal line shows no correlation at all (not a good sign), while a line going from top left to bottom right is a negative correlation (students who did well on one task did poorly on the other: probably undesirable). In general, you would like a positive correlation of the two tasks, but probably you will not see, or even desire, a perfect correlation (a value of 1.0, a slope of 45).

To prevent the linear regression line from being printed, select “Select Show statistics”.

**(b) Lines on chart**

To show or not show lines on a chart, select “**Select Lines on chart**”. The horizontal and vertical lines are shown on the chart to help you line up the scores.

### **(c) Labels**

The chart title, X-axis title, and Y-axis title can be changed by selecting “Select Chart labels”. The default X- and Y-axis titles are the task names, while the chart title is the class name.

Selecting “Output terms” allows changing the statistics names. The output terms affect all parts of the program, while the chart labels affect only this chart. If you change to a different task, the labels will change.

### **(d) Mouse buttons**

The left and right mouse buttons also pop up information.

- Pressing the <Left Mouse Button> while on not on a plotted value will show the correlation coefficient (the slope of the line). If on a value, the student’s name will show. If more than one student have the same value, only one will be shown.
- Pressing the <Right Mouse Button> on a plotted value will pop up the x and y values. Otherwise, the correlation coefficient is shown.

If you print the chart with this data showing on the screen, the data will be printed, as well.

## **5. Printing and copying**

You can print the chart and the data for the chart, and can also copy a chart to the Windows clipboard.

### **(a) Printing a chart**

After you have changed the chart to get it to look how you want, you can print the chart by selecting “Chart Print chart”.

You can select a different chart font. There are five general styles (Roman, Swiss, Modern, Script, and Decorated). Note that the chart fonts are different than the fonts used for printing other data.

### **(b) Printing data for the chart**

The data used to determine the chart can be printed. **VGW** will list the scores in two columns, one for each task. They are sorted by X-axis, then Y-axis values. The

alphabetical rank order of the students is also printed, if that option is selected. In addition, the correlation coefficient is also printed. Select “Chart Print data” to print the data.

The font for the data is the same font used for printing other data. Note that data fonts are different than chart fonts.

### **(c) Copying a chart**

If you select “Chart Copy”, the chart will be copied to the clipboard. You can then paste a chart into any Windows word processor, paint program, and other programs.

## **6. Choosing new tasks**

Selecting “Tasks X-axis task” or “Tasks Y-axis task” will let you choose new tasks to plot.

## **C. Plotting the Scores of an Individual**

You can follow how a student is doing in your class by plotting the scores for that student on a set of tasks. This helps you spot trends in how the student is doing. To plot trends:

- Select “Plot Individual progress”.
- Select the student to plot.
- Select the tasks to plot. The plot will show the tasks plotted using students scores.

You can change way the data is plotted in numerous ways. For example, instead of student’s scores, you can plot the data as percentages, percentiles, standard deviations, T-scores, or student’s t-scores.

### **1. Changing how to plot the data**

The program first plots the data as (raw) student scores. In some cases, the information isn’t meaningful. For example, if some of the tasks are worth 10 points, and others 100, those worth few points will come out very small compared to those worth more points. You can therefore plot the scores with a wide variety of statistical measures. Scores and percentages tell you how well a student did on the task in absolute terms, while the other measures tell you how well the student did in relation to the other students.

To plot with the options below, select “Select”, then the item on the chart menu.

**(a) Scores**

This causes the scores on each task to be plotted. One potential problem with this is that, if each task has a different number of points, it may be difficult to see trends and compare scores.

**(b) Percentages**

This will cause the percentages on each task to be calculated and plotted. This tells you how well a student did on each task, compared to the possible points.

**(c) Percentiles**

The percentiles of the student on each task are calculated and plotted. To calculate percentiles, the student scores are ordered by score, then the position in the class is calculated. A percentile is the percentage of students that a student had better scores than. The best percentile is 99, as that student did better than 99% of the other students. Percentiles therefore rank the students in order of their scores, and tells you the rank of the student compared to the rank of the other students.

**(d) z-scores (standard deviations from the mean)**

The standard deviations (SD) are calculated for each task, and the student's SD plotted. The standard deviation is a measure of how the scores are dispersed. The larger the standard deviation, the greater the range of the scores. A score that is equal to the mean plus the standard deviation (called “one standard deviation above the mean”) is approximately in the top 16% of the scores. For a normal distribution, two thirds of the scores should be within one standard deviation of the mean. How many standard deviations away from the mean a student has scored, therefore, is a measure of how the student did on the task in relation to everyone else. You can compare scores between tests by comparing the deviations from the means for each student. This tells you how a student did relative to the mean score of all the other students.

**(e) T-scores**

T-scores are scores that have been adjusted by the mean and the standard deviation such that the mean of a class is 50, and the standard deviation is 10. Therefore approximately two-thirds of the scores would be from 40 points of 60. T-scores avoid negative numbers and are a little easier to understand than z-scores, but otherwise are comparable.

**(f) Student t-scores**

Student t-scores are a standard statistical measure similar to the standard deviation. However, it does not assume a normal distribution, like a standard deviation does, but instead “samples” the mean. It is a better measure than the standard deviation for small classes, and tells you how a student did compared to the mean score of all the other students.

The program will plot the task scores and calculate the statistics, and will plot the trend and mean lines. The program assumes that the student trend is a straight line: either the scores increase or decrease. Note that this is frequently not the case, hence, these results should be interpreted with caution. The program then calculates the best-fitting line for that trend, and plots it on the graph.

To change whether the trend lines are plotted, select “Select Show statistics”.

## **2. Selecting a different plot style**

There are four general styles of plots. The style can be changed by selecting “Chart”, and then one of the styles.

### **(a) 3D Bars**

This type of plot is a histogram.

### **(b) 2D Bars**

This type is similar to the 3D bars, but the bars look two dimensional instead of three dimensional.

### **(c) Line charts**

Instead of bars, there are lines plotted.

### **(d) Tape graphs**

Instead of bars, there are ribbons (or tapes) plotted.

## **3. Changing the appearance of the chart**

You can change the chart labels, can remove the horizontal and vertical lines on the chart, and can choose to print or not print trend lines on the chart. In addition, the mouse

buttons will also pop up information.

**(a) Statistics**

The program will plot the task scores and calculate the statistics, and will plot the trend and mean lines. The program assumes that the student trend is a straight line: either the scores increase or decrease. Note that this is frequently not the case, hence, these results should be interpreted with caution. The program then calculates the best-fitting line for that trend, and plots it on the graph.

To change whether the trend lines are plotted, select “Select Show statistics”.



### **(b) Lines on chart**

To show or not show lines on a chart, select “Select Lines on chart”. The horizontal and vertical lines are shown on the chart to help you line up the scores.

### **(c) Labels**

The chart title, X-axis title, and Y-axis title can be changed by selecting “Select Chart labels”. The default X- and Y-axis titles are “Task” and the statistical measure, respectively, while the chart title is the student name.

Selecting “View” will allow you to change the way the student name is shown on the chart title.

Selecting “Output terms” allows changing the statistics names. The output terms affect all parts of the program, while the chart labels affect only this chart. If you change to a different task, the labels will change.

### **(d) Mouse buttons**

The left and right mouse buttons also pop up information.

- Pressing the <Left Mouse Button> while on not on a bar will show the value of the trend line (the slope of the line). If on a bar, the student’s value (score, percentage, etc.) will show.
- Pressing the <Right Mouse Button> will show the mean, high, low, and trend values.

If you print the chart with this data showing on the screen, the data will be printed, as well.

## **4. Changing the tasks and student**

If you decide to use a different student or set of tasks, you can change them from the chart.

To change the student, select “Select Student”. Just choose the one you want to plot.

To change the tasks, select “Tasks”, then the ones you want to use.

## 5. Printing and copying

You can print the chart and the data for the chart, and can also copy a chart to the Windows clipboard.

### (a) Printing a chart

After you have changed the chart to get it to look how you want, you can print the chart by selecting “Chart Print chart”.

You can select a different chart font. There are five general styles (Roman, Swiss, Modern, Script, and Decorated). Note that the chart fonts are different than the fonts used for printing other data.

### (b) Printing data for the chart

The data used to determine the chart can be printed. **VGW** will list the tasks and the scores. In addition, the mean and trend is also printed. Select “Chart Print data” to print the data.

The font for the data is the same font used for printing other data. Note that data fonts are different than chart fonts.

### (c) Copying a chart

If you select “Chart Copy”, the chart will be copied to the clipboard. You can then paste a chart into any Windows word processor, paint program, and other programs.

## D. Class and Student Statistics

You can see how a student did in comparison to the rest of your class by plotting the score for that student versus the class. This lets you see graphically how a student did. To do this plot:

- Select “Plot Class vs. Individual”.
- Select the task to plot.
- Select the student to plot. If no student is selected, only the class statistics will be shown.

The plot shows the student plotted on the bottom. The full range of possible scores is the X-axis. The dark wide band shows the range of scores. The mean, median, and standard

deviations (+1 and -1) are also shown. In essence, you have a statistical plot of the class.

## 1. Changing the appearance of the chart

You can change the chart labels, and can remove the statistical measures on the chart. In addition, the mouse buttons will also pop up information.

### (a) Statistics

The range of scores is the dark band on the chart. You can choose not to plot the range by selecting “Select Range on chart”.

The mean, median, and standard deviations can be removed from the chart, as well. Select “Select Show statistics” to turn them on or off.

### (b) Labels

The chart title and X-axis title can be changed by selecting “Select Chart labels”. The default X-axis title is the task name, while the chart title is the class name.

Selecting “View” will allow you to change the way the student name is shown on the chart title.

Selecting “Output terms” allows changing the statistics names. The output terms affect all parts of the program, while the chart labels affect only this chart. If you change to a different task, the labels will change.

### (c) Mouse buttons

The left and right mouse buttons also pop up information.

- Pressing the <Left Mouse Button> will show a variety of statistical measures: the student name and score, the mean, median, standard deviation, the low score, and the high score.
- Pressing the <Right Mouse Button> will show the student name and score.

If you print the chart with this data showing on the screen, the data will be printed, as well.

## 2. Changing the tasks and student

If you decide to use a different student or set of tasks, you can change them from the

chart.

To change the student, select “View Student”. Just choose the one you want to plot.

To change the tasks, select “Tasks”, then the ones you want to use.

### **3. Plotting different students**

In addition to score ranges, there are two additional options that affect which students are plotted. The options listed in parts a and b make a new list of students to plot, while ignoring students just adds or deletes students from the current list of students.

#### **(a) Selecting students**

The options listed here allow you to select any combination of students you wish to plot.

- “Select Students”.
- There are five different options available. You can select no students (not a good idea when plotting), select all the students, select students in sections select individual students, and select students who match a value. These options are explained in more detail in Chapter IX, Section H.

#### **(b) Ignoring students**

Ignoring students prevents them from being used for plots and statistical calculations. Otherwise they are treated like normal students. Ignoring is usually done when a student has a score much different than the other students. This could have resulted from missing assignments or a very good or bad score on an assignment.

- “Select Ignore students”.
- Choose the students to ignore. The students listed are from the current list you have chosen (section, matching values, etc.), so you can refine your choices.

### **4. Printing and copying**

You can print the chart and the data for the chart, and can also copy a chart to the Windows clipboard.

### **(a) Printing a chart**

After you have changed the chart to get it to look how you want, you can print the chart by selecting “Chart Print chart”.

You can select a different chart font. There are five general styles (Roman, Swiss, Modern, Script, and Decorated). Note that the chart fonts are different than the fonts used for printing other data.

### **(b) Printing data for the chart**

The data used to determine the chart can be printed. **VGW** will list the tasks and the scores. In addition, the mean and trend is also printed. Select “Chart Print data” to print the data.

The font for the data is the same font used for printing other data. Note that data fonts are different than chart fonts.

### **(c) Copying a chart**

If you select “Chart Copy”, the chart will be copied to the clipboard. You can then paste a chart into any Windows word processor, paint program, and other programs.

## **E. Calculating and Displaying Statistics**

You can calculate statistics without having to plot the scores. They can either be calculated and printed, or calculated and the values assigned to tasks.

### **1. Explanation of statistical terms**

#### **(a) Mean, median, mode, standard deviation**

The median is the score for the person exactly in the middle of the class, i.e., exactly half the scores are above that number and half are below that number. The mean is the number calculated by adding up all the student scores and dividing by the number of students in the class. If you have a “normal” or “bell-shaped” distribution of scores, the mean and median should be the same number. In practice they are usually different. The mode is the interval with the largest number of students. The standard deviation is a measure of how the scores are dispersed. The larger the standard deviation, the greater the range of the scores. A score that is equal to the mean plus

the standard deviation (called “one standard deviation above the mean”) is approximately in the top 16% of the scores. For a normal distribution, two thirds of the scores should be within one standard deviation of the mean. How many standard deviations away from the mean a student has scored, therefore, is a measure of how the student did on the task in relation to everyone else. You can compare scores between tests by comparing the deviations from the means for each student.

### **(b) Percentiles**

Percentiles are calculated by ranking the students in order of their scores. These ranks are then converted into percentages by dividing the rank by the number of students in the class, subtracting the resulting number from 1.00, and multiplying the result by 100. So, if there are 100 students in the class, and a student is the 30th highest score, you would divide 30 by 100, obtaining 0.3. Next, subtract 0.3 from 1.0. The result, 0.7, is multiplied by 100 to get 70. The student is in the 70th percentile: he or she did better than 70 percent of the students in the class.

### **(c) T-scores**

These are scores scaled by adjusting the mean to be 50 and the standard deviation to 10. They are calculated by determining the mean and standard deviation, then using the student's standard deviation to adjust the scores.

### **(d) Student's t-scores**

Student's t-scores are an estimate of the standard deviation of a population. It is also called a Student's t-distribution. It does not assume a normal distribution, like a standard deviation does, but instead "samples" the mean.

## **2. Displaying statistics**

You can calculate descriptive statistics: the mean, median, mode, standard deviation, and high and low scores are calculated. In addition, a simple non-graphic histogram is shown. You can see these statistics for any combination of tasks.

To calculate statistics:

- Select "Grade Descriptive statistics".
- Select the tasks that you want to use.

### **(a) Altering the window**

- ◇ To view statistics on new tasks, select "Tasks". Then choose the new ones to show.
- ◇ The statistics are calculated for whichever sections you choose. To choose new



sections, select “Sections”.

- ◇ The terms in the window can be changed by selecting “Output terms”.

### **(b) Printing the window**

After the data has been adjusted to look how you desire, it can be printed by selecting “File Print”. The font can also be changed by selecting “File Fonts”. Any Windows font can be used.

## **3. Assigning statistics values to tasks**

Z-scores (standard deviations), T-scores, student’s t-scores, and percentiles tell you how a student did in relation to the other students in the class. You can calculate these statistical measures of a task for the class, and assign these numbers to a number task. To do this:

- Select “Grade Assign statistics values”.
- Select one of the four types of statistics.
- Select the task whose statistics will be calculated.
- Select the new task number. If the task exists, you will be warned that the task will be redefined. If you select “Yes”, the old task will be replaced with the new task.

*Note:* If you assign z-scores (standard deviations) or student’s t-scores, approximately half of the scores will be below zero. Make sure that you change the fewest allowed points to a number below the lowest standard deviation or student’s t-score. A value of -10 or so is usually sufficient.

## **F. Scaling the Tasks**

To scale a task, you add or multiply the score by a number. Therefore, you can only scale number or extra credit tasks. You may add any number of points to an task, or multiply the task by any positive number. The total points for the task, after scaling, will not change. Point totals less than the allowed scores are summed as the lowest allowed score, while scores greater than the total task points are set at the total points. Students with scores that have not been entered are not changed (the score is still undefined).

Scaling allows you to conveniently change point totals for any number task. It also allows

you to add points to one section and not to others, as the scaling occurs only for those sections you choose.

To scale scores:

- Select “Grade Scale a task”.
- Select the task.
- If there is more than one section of students in the class, select the sections to use.
- Select whether to add, subtract, multiply, or divide a certain value. Enter the value.

## **G. Adjusting the Statistics of a Task**

In unusual circumstances, you may want to manually adjust the mean, median, and standard deviation of a task. This is a different way to scale a task. The T-scores described above are an example of adjusting tasks this way. You might want to adjust a task because the scores were too low or high. Adjusting them would make the scores comparable to the others. To adjust statistics:

- Select “Grade Adjust task statistics”.
- Select the task whose statistics need to be adjusted.
- The current mean, median, and standard deviation are shown on the screen. You need to enter the new mean and standard deviation. You can also remove any previous adjustments (if there were any, the previous adjustment box is checked). The student scores are adjusted to reflect the new mean and standard deviation. These numbers are then used by all the other parts of the program that use these values. Note that, if you remove the adjustment, the scores may be slightly different than the true original value due to rounding off errors.
- Select “OK”.

## **H. Selecting students to view**

For most of the plotting and statistics, you can select exactly which students to use for the calculations. The option that allows this is usually “Select Students” or “Select Different students”. There are five different ways you can select students.

## **1. Pick the students**

For this option, you are presented with a list of all the students in the class.

- To add a student, click the <left mouse button> on the student's name in the top list box, then click on "Select".
- To add all the students, click the <left mouse button> on "Select All".
- To reject a student, click the <left mouse button> on the student's name in the bottom list box, then click on "Reject".
- To reject all the students, click the <left mouse button> on "Reject All".
- When you have selected the students you want to use, click the <left mouse button> on "OK".

## **2. Filtering students**

You can also filter students through names, IDs, database items, or tasks. You specify that the students must meet a criterion (above, below, or equal to certain values). For example, if you have a database character field for gender (M or F). You can specify that only males, or only females be plotted.

- Select "Only students who match values".
- Also select "Match Names", "Match IDs", "Match DB items", or "Match Task scores" on the bottom.
- If you selected DB items or task scores, select the one to use.
- Select the value or values and whether to select students above, below, between, or equal to the value or values.

## **3. Use all the students**

All the students in the class will be used.

## **4. Students in sections**

Students in the sections that you select will be used. If there is only one section of students, then all the students will be used.

- To add a section, click the <left mouse button> on a student in the section in the top list box, then click on “Select”.
- To add all the sections, click the <left mouse button> on “Select All”.
- To reject a section, click the <left mouse button> on the section in the bottom list box, then click on “Reject”.
- To reject all the students, click the <left mouse button> on “Reject All”.
- When you have selected the sections you want to use, click the <left mouse button> on “OK”.

## 5. No students

No students will be used. In most cases, this is not a good choice. Many report forms don't need a student to be selected, so this would be a good choice for those forms. See Chapter XI, Section E2d for more details.

## I. Grade Options

These grade options affect calculating and analyzing the grades, and are found by selecting “Task Options” from the Main Menu.

### 1. Fewest allowed points

When you first use the program, the fewest points that a student can have on a task is 0 points. If you want to enter scores below 0, you can set the fewest allowed points to be less than zero. One way that you might have grades below zero is if you assign some of the statistics values to a task. You can reset the fewest allowed points to as few as -1,000,000 points.

Even though the fewest allowed points may be 0 (or any other number), you can still enter scores lower than the fewest allowed points. Students with scores lower than the fewest allowed points will *always* be ignored for statistical tests and plots, and are *always* considered to be equivalent to unentered scores. The difference from unentered scores is that unentered scores can either be ignored or set to 0, while numbers below the fewest allowed points will always be ignored. When you print the numbers, both types of scores

will be blank or print as “Not submitted”. If you set unentered scores to zero, numbers below the fewest allowed points will print as “\*Not submitted” or “\*”. This will let you distinguish between the two types of scores.

## **2. Decimal places**

Task scores are decimal numbers, and are written using decimal numbers. You can specify the number of decimal places to use for printing and calculating grades. You can print from 0 to 5 numbers after the decimal point. The number of decimals to use is specified for regular tasks and for special tasks.

A lower number for the decimals will allow more numbers to be printed on a page. However, distinctions between students are less obvious. Note that special tasks are more likely to have decimal numbers, as they are calculated. For example, averages would most likely have fractions of a number. The number of places specified for special tasks is also used to print statistical measures, like means, medians, and standard deviations.

## **3. Truncating or rounding numbers**

You can choose whether you want numbers to be truncated or rounded. A truncated number just lops off the end of the number, while a rounded number checks the last digit before it removes it, and add one to the last remaining digit if the number is 5 or above.

*Examples:* If you print with 2 places after the decimal point, 11.111 will be truncated to 11.11, and rounded to 11.11. 15.317 will truncated to 15.31 and rounded to 15.32.

Truncating and rounding affect printing data and calculating and assigning grades.

## **4. How to treat unentered scores**

Unentered scores can either be ignored or set to 0. In general, you would want to count them as ignored until the end of the grading period. For scores that you always want to be ignored, enter a score lower than the fewest allowed points. The value will be visible when editing, but not when printing numbers.

## X. ASSIGNING THE GRADES

Just about any grading system can be used with **VAR Grade for Windows**. Each class can have up to four different grading systems of 50 grade categories. This chapter explains how to design your own letter grading system, how to assign grades, and how to print assigned grades. Each letter grade has a name, a value, and a cutoff that is used for assigning grades. Grade values are the numbers used for calculating averages or percentages, while cutoffs are used for assigning scores to letter grades in final tasks.

### A. Changing the Grade Names and Values

When you start a new class, one of the first things you need to do is to change the letter grading system to the grading system you need to use.

The program defaults to the grading system most commonly used in the United States: A through F, with + and - scores. In some other countries, the program defaults to their most common grading system. However, you can define any grading system that has up to 50 different letter grades (plus an “unentered” or “incomplete” grade). Each grade can have a name of up to three characters long. You can also assign point totals to the grades. The grade names and values should be defined and listed in numerical order. You can also have up to four different grading systems for one class. For example, you can have A to F, S and U (satisfactory/unsatisfactory), and P and NP (pass/not pass).

#### 1. Defining new grade names and values

- Select “Grade Grade names and values”.
- Next, you can select a pre-defined grading system or define your own system.

##### (a) Defining your own grading system

There are four options you first need to enter:

- The number of grading systems. You can use 1 to 4 systems. In most cases, you want to use 1 grading system. If you use more than one grading system, you need to define a database item that is used to select the grading system for each student. You are asked which database item to use, and what values are used for which grading system. If a student has a value that doesn't match any of the grading systems, it is assumed the student uses grading system 1.

- Maximum grade value: This is the grade value if the student is perfect. In most cases, it is equal to the highest grade, but in some cases, you might want to enter a higher value. For example, if you use the A to F grading system, you might want to set the maximum value to 100, while an A (the highest grade) might be worth 95.
- Unentered grade: students with unentered grades receive this value. In the US, this is usually called an Incomplete (“I”).
- Use the grade values for grade cutoffs: Grade values and cutoffs are two different things. For example, an A is usually counted as 4.0 for the grade value, but the cutoff needed for an A is usually 90%. The cutoff would be 90. In some cases, you want the two values to coincide. Checking this item sets them as being the same values.

### Multiple grading systems

If you use more than one grading system, you need to select a database item that is to be used for determining which grading system a student uses. You then need to enter the database values to be used for each grading system. See also Section b, below.

### Entering the grade names and values

Next, you enter the grade names and the grade values for the grades in each of the grading systems. Entering names and values is similar to entering grades on the Class Data Entry window.

Keep in mind that the first grading system is the primary one, the others are secondary ones. When plotting letter grades, or doing statistics on letter grades, only the first system is used. For all other parts of the program, the grading system doesn't affect the results.

The grade values you enter are for the grades, not the scores that are needed to obtain the grade. The current grade names and values are listed. Enter your own grades and point values in numerical order. Grade names and point totals can be changed at any time. However, if you have defined any letter or final grades, the new grade names may change the calculated grades. If the number of grade categories has changed, the program may make bad assumptions about which old grade translates to which new grade. You are therefore strongly urged to define the grading system when you make a new class. You can change the grade values at any time without significant problems.

*WARNING: If you change the grading system after assigning letter grades, the new numbers may translate into a different grade than you want.*



## **(b) Multiple grading systems**

When using multiple grading systems for a class, there are two things to keep in mind. First, the first grading system is the primary one. This is the system that is used for plotting and statistics. In addition, it is the system used when the program doesn't match the database item. Second, a database item is used to select the grading system for the student. If no match of the student value and the grading system occurs, the student is assumed to be using grade system 1. Note that the database values you enter must be the same as the student values, so it is best to use simple things like numbers or letters to designate the values. Lower case letters are converted to upper case ones, however.

For example, you can define a "Grade Options" database item, with the values 1, 2, and 3. For each of three grading systems. Then, when you define the multiple grade systems, tell the program to use a value of 1 for the first grade system, etc. If a student has a value of 4, since there are only three systems defined, grading system 1 would be used.

## **2. Mixing letter and number grades**

The previous section described how to assign grade names and point values. This section describes some of the considerations that go into making grade values.

The common definition of an A in the US is to assign it a point value of 4.0. A B is 3.0, and so forth. Some instructors like to assign letter grades to some things, and number grades for others. If you then average the grades, you are averaging a 4.0 with a 100, and getting 52. This is probably not what you want. If you calculate a percentage of a B, you would get a 75%. This also may not be what you want. An alternative is to assign point values on a scale of 100. One possibility is to assign an A a value of 95. Assign a B a value of 85, and so forth. Assign a total of 100 points as the maximum. Averaging a 100 and a 95 would give a 97.5, probably better than the above 52. A percent of a B would give an 85%, again more what you might expect. Another possibility is to assign an A+ = 100, an A = 90, a B = 80, etc. The variations are endless.

## **3. Changing the grade values**

Grade values are the values of letter grades used in the calculations of percentages, averages, etc. They are *not* the values used to assign grades. Those values are the grade cutoffs. The preset values for the letter grades are 4.0 for an A, 3.7 for an A-, etc. You can redefine the grade values at any time and as often as you want.

Every time you do change them, the grades are recalculated to reflect the new letter grade values. Before you change the grade values, you might want to read Section 2, above.

To define grade values only:

- Select “Grade Grade names and values”.
- Select “Change only grade values”.
- Enter the maximum grade value, and whether to use these grade values as cutoffs. See Section 1 for more details about these options.
- Enter the grade values. You cannot change the grade names on this screen.

## **B. Assigning Grade Cutoffs**

Cutoffs are used for assigning final task grades. Note that these are different from grade values. We recommend that you define the grade cutoffs before you define any Final tasks. For instance, you can tell the program that 90% is an A, 80% is a B, etc. Alternatively, you can tell the program that 90 points is an A, 80 points a B, etc. When you assign grades, **VAR Grade for Windows** will use these values, either as percentages or points, as the point system to use when assigning grades. You can change these cutoffs at any time. However, when Final tasks are defined, they inherit the current cutoffs. If you change the cutoffs after the Final task is defined, the task definition keeps the old cutoffs. To use new cutoffs, you would have to reassign grades by selecting “Grade Assign final grades”.

### **1. Assigning cutoffs**

- Select “Grade Grade cutoffs”, then “Assign cutoffs”.
- There are two options that you can set: First, you need to select whether the cutoffs are percentages or absolute numbers (total points). If you pick percentages, the total points of each task are multiplied by the percentage to obtain the cutoffs. If you pick points, no matter how many points the task is worth, the student will need to obtain these points to get the grade. In most cases, you want to define the cutoffs as percentages. Second, you can select whether to make the grade values the same as the grade cutoffs. See Section A for more details on this option.
- You next enter the minimum scores (percentage or points) to obtain each grade. The cutoffs must be listed in numerical order. If you leave a cutoff blank, no students will obtain that particular grade. If you use multiple grading systems,

each system is listed separately.

## 2. Considerations

If you define cutoffs before defining a Final task, the values of the cutoffs are used to compute the grades. If no cutoffs have been assigned, you must manually assign cutoffs for the final task. To remind you, if you try to define a Final task before assigning cutoffs, you are asked whether to assign the cutoffs right then.

If you change or remove the cutoffs, this will not change the grades in any final tasks that have already been defined, but any final tasks defined after you change the cutoffs will use those new cutoffs.

To change the cutoffs of Final tasks that have already been defined, select “Grade Assign final grades”.

## 3. Removing cutoffs

To remove the cutoffs, select “Grade Grade cutoffs”, then “Remove cutoffs”. This does not remove the cutoffs for Final tasks that have already been defined.

## C. What Ifs

“What ifs” are calculations of what a student needs to obtain a particular grade. Students frequently ask instructors what grade they need to get a grade: **VAR Grade for Windows** can answer their questions. However, they are calculated on Final tasks. You therefore must have defined the Final task before doing any what ifs.

What ifs will calculate the *average* grade(s) needed by a student to achieve any and all letter grades for a final task. Sometimes an exact solution is possible (e.g., when only one score hasn't been entered). Sometimes the score must be estimated (e.g., when you discard more than one task). In either case, the score is an average for the missing scores. When scores are discarded, it may be possible for students to get below the average and still get the desired grade. Therefore, the calculated score can be regarded as the average score that guarantees the grade, not the minimum average needed to achieve a grade.

To calculate What ifs:

- Select “Grade What ifs”.

- Select the Final task to use.
- Select the student(s) that want the what ifs.
- If grade cutoffs have not been assigned, you are asked for them before the what ifs can be calculated.

The what if printout contains a lot of information: the current points for the student, the total points for tasks that have been taken, the points left to be recorded, and the maximum points possible. Each grade is listed, along with the points or score needed for the grade (“Min. score for grade”), and how much the student needs on the remaining tasks to achieve each grade (“Needed scores”). If the student has already achieved a grade, that is also indicated, or, if the student cannot obtain that grade, that is also indicated. The percentage of the remaining points needed for each grade is also listed (“Pct. needed”).

### **1. Changing the cutoffs**

If you select “Cutoffs” from the What if Menu, you can change the cutoffs for the task. After you enter the new values, you will be asked whether to make the cutoffs permanent for that task. If you select “Yes”, the grades will be recalculated for the students in the class, otherwise the values only affect that student for that what if session, as moving to another student would cause the cutoffs to revert to the original ones.

### **2. Changing the appearance**

Selecting “View” will allow you to change the way the student name and ID is shown with the data.

Selecting “Output terms” allows changing the what if titles. The output terms affect all parts of the program.

### **3. Changing the task or student(s)**

To change the students, select “Students”. To change the task, select “Tasks”. These options give you the chance to calculate what ifs on different students in the same session, and for different tasks for the same student in that session.

### **4. Printing the results**

Select “File Print” to print the what ifs. The font for the data is the same font used for printing other data, and can be changed by selecting “File Fonts”.

## **D. Assigning Final Grades**

Final tasks inherit the grade cutoffs that have previously been defined. If no cutoffs had been defined, all grades are the highest, or incomplete. You can assign or change the cutoffs for a specific Final task after it has been defined:

- Select “Grade Assign final grades”.
- Pick the task to grade. Final tasks will have the grades assigned to it, while others are only temporarily assigned.
- Choose the new cutoffs. If ones have previously been assigned, the values are shown.

The program lists the points, percentages, and assigned grades, by section, for the task. A tables of numbers of students with each grade and the cutoffs are then listed. Only grades that students can earn are listed at the end of the class: if the cutoff is blank, it will not be listed.

### **1. Assigning new cutoffs**

Changing the cutoffs will probably be a significant change you will make when assigning the grades. You can “tweek” the grades until you are satisfied with them.

If you select “Cutoffs” from the Assigned grades menu, you can change the cutoffs for the task. If the task is a Final task, the changes are permanent, otherwise they are temporary.

### **2. Changing the appearance**

Selecting “View” will allow you to change the way the student name and ID is shown with the data.

Selecting “Output terms” allows changing the data titles. The output terms affect all parts of the program.

### **3. Printing the grades**

Select “File Print” to print the grades. The font for the data is the same font used for printing other data, and can be changed by selecting “File Fonts”.

The printout contains the assigned grades, the numbers of students assigned to each grade category, and the average grade that was assigned. For each category, it also print the percent of the total students who received that grade.

## **E. Adjusting Grades**

If you do not like the grades that have been assigned, you can adjust them. This is frequently done when students have extenuating circumstances that have adversely affected their grade. In fact, the major purpose of adjusting grades is after all the grades have been entered, analyzed, and assigned. Anomalies and other adjustments can then be made. Adjustments can only be done on Final tasks. To adjust grades:

- Select “Grade Adjust grades”.
- Select the task to adjust. The task must be a Final task.
- Both the final grades and the scores used to compute them are listed, each in it's own column. You can adjust either or both scores. The data is edited exactly like editing in the Class Data Entry window. The major difference is that some options are not available to you.
- When finished, select “Exit”.

The scores are then saved as *new* tasks (with new names: “Adjusted” is prepended). These new tasks are defined as letter or number tasks, whichever is appropriate. Although you could do these adjustments manually, this is much simpler. If you don't like the names to have “Adjusted” in them, you can change the names by selecting “Task Alter or move”, then “Rename”.

## XI. PRINTING AND REPORTS

**VAR Grade for Windows** can print student data almost any way you want. There are a wide variety of ways the program will print the data that just requires you to select the data and print it. There is also a report writing section that lets you specify exactly where each piece of data should be printed. The reports require more effort on your part, but allow you to present the data almost any way you would like.

### A. Printing header information

When data is printed information identifying the data is first printed. These include the instructor's name, date, class name, and sections. In addition, other identifying information may also be printed. These are discussed below.

#### (a) Class name

When **VAR Grade for Windows** prints data, it also prints the name of the class. Renaming the class will override the default name (which is the first part of the filename). To change the class name on the printouts:

- Select "Write Class name for printouts".
- Enter a new name of up to 30 characters.

*Example:* The file name of a class is called "BIO.PAR". The default name for printing would be "BIO". You can change it to anything else, e.g. "Biology 101", for printing purposes, but the file name(s) will still be BIO (plus the extensions ".PAR" and ".DAT").

#### (b) Instructor name

You can list your name when data is printed. If a name is entered, it is printed when data is printed. Otherwise, no space for an instructor's name is printed. To change the instructor's name:

- Select "Write Teacher name".
- Enter your name.

#### (c) Section names or comments

If you use more than one section of students, you can specify comments or names for the section on printouts. The comment is written instead of the section number. If there is only one section of students, the comment is not printed, as the section number is not printed.

- Select “Write Section name or comments”.
- Enter a name for each section in your class. This will replace printing “Section x”, where x is the section number.

*Example:* You can have the program write “Section 1: MWF 9 to 10 AM” instead of the section number on printouts.

#### **(d) Date**

The current date is written when data is printed. If you have failed to set your clock, the date will not be correct. You can change the format of the dates by selecting “Option Date format”. The date format also affects how attendance days are defined by the program. You can also change the names of the day of the week and the months. Select “Option Date names”. See Chapter VIII, Section A for more details on changing the dates.

## **B. Printing the Student Scores and Attendance**

After entering data, you probably want to print it. There are numerous ways to print data in **VAR Grade for Windows**.

- You can print data for individual students by selecting “Write by Name”.
- You can print data for whole sections of students by selecting “Write by Section or class”.
- You can write empty grids for entering student data by selecting “Write Empty grids”.
- You can design and print your own reports. See section D.

For the first two, you can print task scores and attendance a variety of ways.

### **1. Writing data for individual students**



You can print data from one individual a variety of ways, including task scores, attendance, missing assignments, missing attendance days, and summaries of attendance. In each case, the data is printed as one student per page (or pages, if needed). You print the data with:

- Select “Write by Name”.
- A variety of options is available:

<u>Select</u>	<u>Action</u>
“Tasks”	Prints selected task scores.
“Missed assignments”	Prints missing task scores.
“Attendance”, “Attendance”	Prints selected attendance.
“Attendance”, “Missing days”	Prints days the student was absent.
“Attendance”, “Summary”	Prints the totals for each attendance type.
“Attendance”, “Tasks and attendance”	Prints both tasks and attendance.
“Attendance”, “Tasks and summary”	Prints both tasks and summaries.

- Select the student or students to view.
- For most options, select the tasks or attendance days to use.

You now see the data. You can select the various menu options to change the data that is viewed, or how the data is presented.

#### **(a) Viewing tasks**

When viewing tasks, the data is written with the task names on the left, then the student scores, the possible points on an assignment, and then the percentage score.

Note that if a score is unentered, it will be printed as “Not submitted”. You can change that wording by selecting “Output terms”, then “Attend and task terms”.

You can also print database items on the printout. Select “DB” to add them to the printout. They will be printed before the tasks.

#### **(b) Viewing missing assignments**

Selecting “Missing assignments” will print only those tasks where a student has an unentered score. The tasks will be printed by task name, not the number of the task.

#### **(c) Viewing attendance**

Attendance is printed like the tasks: the dates on the left and the attendance on the right. The attendance is printed by the long name (description), so, if a student was present, it is printed as "Present". You can change the names of the attendance types by selecting "Attendance Options" from the Main Menu.

#### **(d) Viewing missing days**

You can print all the days a student was missing from class. Missing days include those that are unexcused, unexcused 2, other 1, other 2, and not entered.

#### **(e) Viewing attendance summaries**

Attendance summaries total up the scores of each attendance type for the student. Each of the attendance categories is printed if the student has at least one of that category. Since there are up to 13 categories, there are up to that many attendance types printed. In addition, the attendance is totaled into one of three major categories: days present, days absent, and days not recorded. Days present are the attendance types present, present 2, late, late 2, activity 1, activity 2, and activity 3. Days absent are excused, excused 2, unexcused, unexcused 2, other, and other 2. The other category is days not recorded.

#### **(f) Viewing database items on the printouts**

Database items can be printed on the individual printouts. They are printed after the name, ID, and section, but before the tasks or attendance. There are two ways to add database items onto printouts:

- Select “Write by Name”, then “DB items on printouts”. Then select the items you want printed, and in the order you want them printed.
- Select “Write by Name”, then one of the seven printing types described above. After the data is shown, select “DB”. Select the items in the order you want them printed.

The items selected will continue to be printed until you select a new list.

#### **(g) Viewing other classes or subjects**

Elementary school teachers usually teach several subjects to the same set of students. Although in most cases you want to print data from only one class, VGW has the capability to print data from as many as 10 classes in addition to the current one on the same printout. This is useful when printing consolidated reports of students who are taking more than one subject from you. To print more than one class, there are two methods:

- Select “Write by Name”, then “Extra subjects or classes”.

- Select “Write by Name”, then one of the seven printing types described above. After the data is shown, select “Subjects”.

### I. Adding new subjects

There are 10 spaces for the subjects. To add a new subject, there are four methods:

- Move to an empty field, then type in the class file name. This must include the full DOS path to the file (e.g., “C:\VARGRADE\CLASS”).
- Move to a field with a class already listed. Delete the old class, then type in the class file name. This must include the full DOS path to the file (e.g., “C:\ENGLISH\ENG101”). This method replaces the old subject with a new one to be printed.
- Move to an empty field, then press the “Add new subject” button. Then use the Open Class Dialog to find and select the class.
- Move to a field with a class already listed. Press the “Replace subject” button. Then use the Open Class Dialog to find and select the class. The old subject on the list is replaced with the one you select. Note that if you press the “Add new subject” button instead of the replace button, the new class will be added to the first empty field.

### II. Removing subjects

There are two methods to delete a subject:

- Move to the subject to delete. Press the <Backspace> key until the name has disappeared.
- Move to the subject to delete. Press the “Delete subject” button.

After you have added and/or changed the subjects, select the “OK” button. When the program prints the student data, it will cycle through the classes in order, starting with the current class, and print the data. If necessary, it will ask for the tasks or attendance days to use for printing.

### **(h) Changing the appearance**

Besides having numerous ways to print the data, you can also change parts of the

appearance of the printouts.

Selecting “View” will allow you to change the way the student name and ID is shown with the data.

Selecting “Output terms” allows changing the data titles. The output terms affect all parts of the program.

Selecting “File Options” will let you change more options. These are described in Section C.

### **(i) Printing the data**

Before printing, you can change the appearance, tasks, students, attendance, subjects, and database items until the data looks how you want. You can then print by selecting “File Print”. The font for the data is the same font used for printing other data, and can be changed by selecting “File Fonts”.

## **2. Writing data for sections or the entire class**

Besides writing data for an individual student, you can print data from any combination of sections, including the entire class, a variety of ways. These include task scores, attendance, missing assignments, missing attendance days, and summaries of attendance. You print the data with:

- Select “Write by Section or class”.
- A variety of options is available:

<u>Select</u>	<u>Action</u>
“Tasks”	Prints selected task scores.
“Missed assignments”	Prints missing task scores.
“DB items only”	Prints database items.
“Attendance”, “Attendance”	Prints selected attendance.
“Attendance”, “Missing days”	Prints days the student was absent.
“Attendance”, “Summary”	Prints the student totals for each attendance type.
“Attendance”, “Brief summary”	Prints the present and absent totals for students.
“Attendance”, “Totals”	Prints the totals by class or section for each attendance type.

These items can also be selected when the data is shown. Select a new item on the “Tasks”, “Attendance”, or “DB” menu choices.

- If you have more than one section of students, select the sections to print.
- For most options, select the tasks or attendance days to use.

You now see the data. You can select the various menu options to change the data that is viewed, or how the data is presented.

*NOTE: If there are too many items to print on one screen, the program will print out as many as fits across the page for a whole section, then another set, etc., until all the items and students have been printed.*

#### **(a) Viewing tasks**

When viewing tasks, the data is written with the student names on the left, then the student scores in columns. If there are more tasks than can be listed in one part, the program breaks the list into parts. Each section is printed separately. Note that if a score is unentered, it will be printed as a blank. Excused scores, those with fewer points than the fewest allowed points, are printed with an asterisk (“\*”). You can print the data with a key at the top or without one. Keys let you print more tasks per page, while making the printouts less readable. This option can be changed by selecting “File Options”.

#### **(b) Viewing missing assignments**

Selecting “Missing assignments” will print only those tasks where a student has an unentered score. The tasks will be printed by task name, not the number of the task.

#### **(c) Viewing attendance**

Attendance is printed like the tasks: the names on the left and the attendance in columns on the right. The attendance is printed with the attendance characters you specify. To change them, select “Attendance Options” on the Main Menu.

#### **(d) Viewing missing days**

You can print all the days students were missing from classes. Missing days include those that are unexcused, unexcused 2, other 1, other 2, and not entered.

### **(e) Viewing attendance summaries**

Attendance summaries total up the scores of each attendance type for the student. Each of the attendance categories is printed if the student has at least one of that category. Since there are up to 13 categories, there are up to that many attendance types printed. In addition, the attendance is totaled into one of three major categories: days present, days absent, and days not recorded. Days present are the attendance types present, present 2, late, late 2, activity 1, activity 2, and activity 3. Days absent are excused, excused 2, unexcused, unexcused 2, other, and other 2. The other category is days not recorded. The printing order is to print the days present, then each present category with at least one student, the days absent, then each absent category with at least one student, then the days not recorded.

The “Brief summary” lists the totals, but not the numbers for each category, so it would list only “Days present”, “Days absent”, and “Days not entered”. Note that the titles can be changed by selecting “Output terms”.

### **(f) Viewing attendance totals**

Attendance totals are summaries of the numbers of students in each attendance category. These differ from the attendance summaries in that they total the students, and do not print each student.

When you total the days, you can select whether to total the days separately or together. If done separately, a similar total would be done for each day.

### **(g) Printing database items**

Database items can be printed. The items are listed like attendance or tasks, with the names on the left, and the items on the right in columns. In the case of memos or the longer strings, the item may print on more than one line.

### **(h) Filtering through database items**

Database items can also be used to filter students. Those that meet the criterion you select will be printed, others will not be printed. Any database item can be used.

To select database items to filter, there are two methods:

- Select “Write by Section or class” on the Main Menu, then “Filter through db items”. You then select the database item to use for the criterion. You then use a dialog to enter the filter criterion.

- Select “DB DB Filter” when viewing the data.

#### **(i) Viewing other classes or subjects**

Although in most cases you want to print data from only one class, VGW has the capability to print data from as many as 10 classes in addition to the current one on the same printout. This is useful when printing consolidated reports of students who are taking more than one subject from you. To print more than one class, there are two methods:

- Select “Write by Section or class”, then “Extra subjects or classes”.
- Select “Write by Section or class”, then one of the printing types described above. After the data is shown, select “Subjects”.

Section 1g, above, has details on how to add or remove classes. When the program prints the student data, it will cycle through the classes in order, starting with the current class, and print the data. If necessary, it will ask for the tasks or attendance days to use for printing.

#### **(j) Changing the appearance**

Besides having numerous ways to print the data, you can also change parts of the appearance of the printouts.

Selecting “View” will allow you to change the way the student name and ID is shown with the data.

Selecting “Output terms” allows changing the data titles. The output terms affect all parts of the program.

Selecting “File Options” will let you change more options. These are described in Section C.

#### **(k) Printing the data**

Before printing, you can change the appearance, tasks, students, attendance, subjects, and database items until the data looks how you want. You can then print by selecting “File Print”. The font for the data is the same font used for printing other data, and can be changed by selecting “File Fonts”.



## **4. Printing grids**

For entering student data, it is often useful to list the students with a blank grid. You can then use it to record the grades. The grid can make it easier to enter student grades into the computer, as it's organized the same way as the data on the computer. The grid size is selectable by you, so you can enter small data like checks, or long data like comments.

To print empty grids, select "Write Empty grids". You now see the grids, and can alter the appearance or print it.

### **(a) Changing the box sizes**

To change the width of the boxes, select "Box size". The width can be from 1 to 20 characters wide.

### **(b) Changing the appearance**

Selecting "View" will allow you to change the way the student name and ID is shown with the data.

Selecting "Output terms" allows changing the data titles. The output terms affect all parts of the program.

Selecting "File Options" will let you change more options. These are described in Section C.

### **(c) Printing the data**

You can print the grids by selecting "File Print". The font for the data is the same font used for printing other data, and can be changed by selecting "File Fonts".

## **C. Write options**

There are several options that you can change that will affect how the data is printed. These are on the Write Options Dialog. This dialog can be reached by selecting "Write Options" from the Main Menu, or "File Options" from the menus shown for printing by name, by section or class, or printing grids.

### **1. Print student numbers**

You can change whether numbers are printed next to the student names when listing students by section. The numbers listed are the sorted order for the students. If you sort the students one way, and print them another, you may find that numbers are skipped or out of order.

## **2. Print Keys**

You can choose whether to print a key of tasks, database items, or attendance names at the top of the page, then list the items by number, or whether to print the name above the columns of data. Printing keys will allow more data to be printed per page, however, not printing them makes the printout easier to read.

## **3. One set per page**

You can select whether to print one section or set of data is printed on a page, or whether to try to fit as much data on a page as possible. One set per page will make the printouts more uniform, while fitting as much data as possible on a page will save paper.

#### **4. Truncate vs. rounding**

Truncated scores drop the least significant digit, while rounded scores look at the least significant digit and increase the previous digit if the least significant number is a 5 or greater. For example, 85.55 will round to 85.6 and truncate to 85.5. Note that **VGW** uses the number of decimal places specified by you to determine which digit to round or truncate. If you set the decimals to 1, you get the above rounding. If you set the decimals to 0, 85.55 rounds to 86 and truncates to 85.

Rounding vs. truncating affects how grades are determined. For example, if you require 90% for an A, and round, a student who has 89.5 could receive an A, while that student wouldn't receive an A if the score is truncated. In most cases, people usually truncate scores.

#### **5. Hash character**

The hash character (“#”) can be changed to any other character. This is included for users outside the USA.

#### **6. Minimum allowed points**

This value is the minimum points for a valid student score. If you enter a score lower than that value, the score is always ignored. This lets you excuse students from assignments.

#### **7. Maximum column width on printouts**

This value affects the width of columns when entering data and on printouts. **VGW** determines a width from the larger of the column title or the width of the data. If you enter a number less than the column title width, the title will be split, if possible, into two or more lines. This allows you to have the titles above the columns instead of numbers, and yet still have lots of columns per width of the page.

#### **8. Decimals**

You can choose how many decimals to use for printing and grade calculations. See truncate vs. round for information on how decimals affect grades. For printing, the fewer the decimals, the more data you can have on a page or screen. The drawback is a loss of accuracy in printing the numbers (since they must be truncated or rounded). Changing these values does not affect the accuracy of the grade stored for the student, but just the

printing accuracy.

You can change the decimals for “regular” and “special” tasks. Regular tasks are the ones you enter data from, and, for decimals, Sum tasks. Special tasks are any other tasks. These are calculated and therefore are likely to have decimals in the numbers. The default values for each are 1. If you never have scores that are fractions, you probably want to set the regular decimals value to 0.

## D. Exporting Data

You can export data from a **VAR Grade for Windows** class in two different formats that other programs, like spreadsheet and database programs, can then read import the data. **VGW** can also read the exported data (see Chapter IV, Section D). In addition, you can export data directly to a **VGW** class.

For all the methods of export, you can specify whether to write names, IDs, and sections, as well as database items and any combination of tasks.

The two export methods that other programs can use are comma and quote files and flat files. Comma and quote files are files that have the data separated by commas. Any data that has commas in it is surrounded by quotes. Flat files are one where the data is in columns, like a spreadsheet. The columns can be separated by spaces or <tabs>.

### 1. Comma and quote file

A quote and comma file is a plain text (ASCII) file where strings (names) are enclosed in double quotes (“”), while numbers are not. Cells of data are separated by commas. Many spreadsheets and database programs can import this type of file. Each set of data (or cell) will be imported into one cell in a spreadsheet or database file. Each line in the file ends up as one line in the spreadsheet.

*Example:* a comma and quote file would look like the following:

```
"Darwin, Charles", 10.3, 20, "A", 30  
"Crick, Francis", 20, 15.5, "B", 35
```

In the above example, the file has two students, the first set of data is their names, the second through fifth are grades. The fourth column is letter grades, the other grade columns are number grades.

The first line of an exported comma and quote file contains the names of the student data.

For example, if you exported student names, the first column on the first line will read “Name”. If the data is a task that uses numbers, the name will include the total points of the task.

To export data to a comma and quote file:

- Select “File Export”, then “Comma and quote”.
- Select the database items to export.
- Select the tasks to export.
- Select how to export names and IDs.
- Select the name of the file for export. An extension of “TXT” is suggested, as most other programs assume comma and quote data is in a file with that extension.

## 2. Flat file

A flat file is a plain text (ASCII) file where data is separated into columns. The columns are separated by spaces. Many spreadsheets and database programs can import data in this format. Each set of data (or cell) will be imported into one cell in a spreadsheet or database file. Each line in the file ends up as one line in the spreadsheet.

Example: a flat file would look like the following:

Darwin, Charles	10.3	20	A	30
Crick, Francis	20	15.5	B	35

In the above example, the file has two students, the first set of data is their names, the second through fifth are grades. The fourth column is letter grades, the other grade columns are number grades.

The first line of an exported comma and quote file contains the names of the student data. For example, if you exported student names, the first column on the first line will read “Name”. If the data is a task that uses numbers, the name will include the total points of the task.

When you export flat files, you are asked whether to separate the fields with <Spaces> or <Tabs>. Some programs require data to be separated by <Tabs>, while others don't care. If given a choice, use <Tab> separated data, as the first and last names of the students

could end up in different columns if data is separated by spaces. Whether this happens or not depends upon the program used to import the data.

To export data to a flat file:

- Select “File Export”, then “Flat file”.
- Select the database items to export.
- Select the tasks to export.
- Select how to export names and IDs.
- Select the name of the file for export. An extension of “TXT” is suggested, as most other programs assume flat file data is in a file with that extension.

### **3. Exporting data to Paradox for Windows**

Below is an example of how to export data, then import it into Paradox for Windows. Importing into other databases or programs may work differently, but the general idea is the same.

#### **(a) Export as a comma and quote**

First, export your class data, as explained in part 1, as a comma and quote file. You will need to select a name for the file. Any name will do, but write down the name.

- Select “File Export”, then “Comma and quote”.
- Select the database items to export.
- Select the tasks to export.
- Select how to export names and IDs.
- Select the name of the file for export. An extension of “TXT” is suggested, as Paradox for Windows prefers that extension. Write down the file name and directory.

The data has now been written. Next, you need to start up Paradox for Windows.

### **(b) Import into Paradox for Windows**

The following steps explain how to import the data into Paradox for Windows. It is assumed that you know the basics about the program.

- Select “File Utilities”, then “Import”.
- On the File Import Dialog, select a file type of <Delimited text>, and the name jotted down when exporting the data. You may need to use the “Browse” button to find the file.
- Select a new table name, and make sure, in the Options Dialog, that “comma” and “quotes” are selected.

Paradox now imports the table. The first line of the table will have the names of the items (Name, ID, etc.). The other lines have the students and their data.

## **4. Exporting data to Quattro Pro for Windows**

First, export to a flat file exactly as described below.

- Select “File Export”, then “Flat file”.
- Select data separated by “Tabs”.
- Select the database items to export.
- Select the tasks to export.
- Select how to export names and IDs.
- Select the name of the file for export. An extension of “TXT” is suggested, as Quattro Pro recognizes this extension. Write down the file name and directory.

Next, follow the steps below to import the data into Quattro Pro for Windows. Other spreadsheet programs may work differently. Start Quattro Pro for Windows.

- Select “File Open”.
- Select the file you exported.

The data is then read into the program. The first line of the data will have the names of the items (Name, ID, etc.). The other lines have the students and their data.

## 5. Exporting data to Microsoft Works

This method is almost exactly like that for Quattro Pro for Windows. First, export your data as a comma and quote or a flat file as described above.

Next, follow the steps below to import the data into Microsoft Works. Other spreadsheet programs may work differently. Start Microsoft Works.

- Select "File Open existing file".
- Select the file you exported. It's easier to find the file if you change the file listing on the bottom of the dialog to "\*.txt".

Next, select whether to open the file as a database or a spreadsheet file. In either case, the data is read and the data put into columns and rows. The first line of the data will have the names of the items (Name, ID, etc.). The other lines have the students and their data.

## E. Reports

A simple programming language, named VRL (**VAR Grade Report Language**) has been added to **VAR Grade for Windows** so that you can write reports that include your class data. The language will allow you to write just about any report that you might need. The reports are written in plain text (ASCII characters) that are called report forms. These forms are translated by **VGW** into complete reports. Included with the program are forms that can be used by you for your classes, or used as examples to help you write your own reports. The language used for the forms can be used to emulate the types of output that **VGW** already does. It also lets you write reports that can include data on students from more than one class, so that you can have the program write reports that will consolidate the grades for your students that are taking more than one of your classes.

### 1. Overview of the VRL language

The purpose of VRL is to let you access just about any piece of data of your class. It is also a forgiving language, in that if you make a mistake, the program will not blow up or go into computer heaven. Instead, mistakes will show up as reports that do not look how you want them, and will usually result in the incorrect commands appearing in square brackets in the output.



Each command in the VRL language starts with a left square bracket "[", and ends with a right square bracket "]". The only exception to this are non-printing commands, which start with a left curly bracket "{" and end with a right curly bracket "}". The form is read line by line. This means that commands should be entirely on one line. You can have as many commands on one line as will fit. Each line in the form can have up to 1024 characters, with no limit on the number of lines.

The commands can be broken down into several categories.

- **Comments:** The comments are ignored by **VGW**, but are useful for commenting on what you have done in case you have to later modify the form.
- **Printer and format commands:** Printer commands tell the printer to do something, such as print the next part with bold type. Format commands tell the program to change the way things are written, such as to go to the next page (called a "formfeed").
- **Commands that let you write class data:** These let you write student, task, and attendance data.
- **Commands that tell the program to change something:** These are used to change or switch data, like changing the class, to switch to a different task, and so forth.
- **Commands that tell the program to repeat things:** These allow you to easily write lists of students and tasks.
- **Plain text:** Anything that is not in brackets is written exactly as it is in the report form.

Commands can also be formatted. This means that you can specify not only to write a particular score or name, but also how to write it (spaces, how many decimals, etc.). Different commands need different types of formatting, so the relevant formatting will be discussed with each command.

Commands need to have square brackets around them. Subcommands, like those in the Print command, need to have both the command and the subcommand between the square brackets. Although spelling is important for the commands, they can be in any mixture of upper and lower case letters.

If you want to write a left square bracket in your text, you can do so by writing two consecutive square brackets ("[["). Extraneous material that is written after the

commands, but inside the brackets, is usually ignored.\_

## 2. Writing and viewing report forms

To write report forms:

- Read the following sections on what the commands do. This will let you know what types of things you can do, as well as what some of the funny looking commands mean.
- Browse through some or all of the forms included with the program. This will help you see how you can accomplish what you want to do. Feel free to copy, modify, and discard anything in the forms.
- Write the form in plain text (also called “ASCII” characters). Most, if not all word processors and all text editors can write what you have written in plain text. Word processors usually write files with their own proprietary formatting. To write files in plain text, you generally have to specifically tell the program to do so. If your word processor cannot print plain text files, there is a simple word processor included with **VGW** that writes files only in plain text. To use it, select “Write Reports”, then “Edit a form”. There is more information on the editor in Section a.

### (a) Editing forms

To use the editor in **VGW**:

- Select “Write Reports”, then “Edit a form”.
- Select a name for the file. An extension of “.FRM” is suggested, but any name can be used. The forms included with **VGW** all have the “.FRM” extension. If you use a consistent ending for the forms, it will be easier to find the one that you want.

Edit the form. The editor can do simple editing, including the standard cutting, pasting, and searching. You can also print the form. If you want to view the form while editing, you must save it by selecting “File Save”, then select “Write Reports” from the Main Menu of **VGW**, then “Show a report”, then select the file.

### (b) Adding extra classes

Normally, only the current class is printed with a form. You can, however, print data from up to 10 classes with a single form. Note that there is also a command that can be added to reports to open and print a new class.

- Select “Write Reports”, then “Classes or subjects to list”.
- Add or delete classes from the list. For more details on how to do this, see Section B1g.

### **(c) Automatic comments**

Automatic comments are comments that can be printed for students if they have achieved certain grade cutoffs. You must specify the command [Comment] on the form for the comment to print. The program then selects the comment for the appropriate grade.

Enter automatic comments by:

- Select “Write Report”, then “Add Automatic comments”.
- You can type in the comment and the minimum score needed for that cutoff for up to 9 items. The minimum scores are absolute numbers needed for the comment, not the percentages. If you use the command [Comment?] in a form, the program will ask you to type in the comment when the form is shown. You either pick one of the automatic comments, or type in your own comment. You can also print blank or underlined lines and write them on by hand. The report form Progress.FRM shows the use of the [Comment?] command.

### **(d) Reports**

You can write forms that intend to list most or all of the students in your class, or ones that write the data for only a single student. The former are called “Class forms”, the latter “Individual forms”. **VGW** makes no distinction between the two types, but some commands are usually used in one or the other type of form.

To view report forms:

- Select “Write Reports”, then “Show a report”.
- Select the name of the report form.

Examples of Class forms are listed below. At the end of the report writing section there are examples of how the printouts for these forms look.

<u>Form</u>	<u>Use</u>
Attend.FRM	Prints attendance totals for all the students in the class.
Consolid.FRM	Lists all the students grades on one task in each of three classes plus the current class. The classes and tasks are requested when the form is run. This shows the capability of the report writer in being able to find and write the grades of students (having identical names) from different classes.
Flat.FRM	Emulates flat files. This writes all the students' names, IDs, and sections as a flat file.
Filter.FRM	Shows how to use filters to print out students below/above certain values on task scores and database items.
MailAll.FRM	Mail list that prints the addresses of all students in a class. For the form to work, the fields must be "Street 1", "Street 2", "City", "State", and "Zip". You can change these database names to fit your classes.
NoLine.FRM	Illustrates the use of the [NoLine] command by printing the students or tasks on one line.
Quote.FRM	Emulates quote and comma files. This writes all the students' names, IDs, and sections as a quote and comma file.
<u>Form</u>	<u>Use</u>
Section.FRM	Lists all the grades for all the students in the class. This emulates the "Write students in sections" option of <b>VGW</b> .
Section1.FRM	Lists all the grades for all the students in the class. This form is similar to Section.FRM, but only prints out one task score.
Stats.FRM	Lists the total points, high, low, mean, median, and standard deviation for the first five tasks in a class.
Taskdef.FRM	Lists all the task definitions in the class.
Update.FRM	Lists the name, ID, and one grade for all the students in the class. It also lists all absences.

Forms intended to print individuals are listed below:

<u>Form</u>	<u>Use</u>
Attexamp.FRM	Prints many of the attendance commands.
Classes.FRM	Lists the grades for a student from one class.
Database.FRM	Uses some of the database capabilities to print a form.
Elem.FRM	Prints grades from several classes for one student. This shows how to write a report that consolidates the class data for one student into one report. This type of form is frequently needed for elementary schools.
Low.FRM	This is an example of a report that will tell the parents of students that their child is not doing well enough in your class.
Mail.FRM	Form that prints out a sample mail list. It is different from MailAll.FRM in that it prints out only selected students.
Name.FRM	Lists all the grades and data for one student. This emulates the "Write data for a student" option in <b>VGW</b> .
Namefont.FRM	The same as Name, but illustrates changing fonts.
Progress.FRM	An example of a progress form that can be written. It uses different font sizes.
RCElem.FRM	Form that prints out a sample reportcard for a student. It includes the student's average on the listed tasks.
<u>Form</u>	<u>Use</u>
Report.FRM	This shows the capability of writing a free form report that tells the students in your class how they are doing.
Rprtcard.FRM	This form shows possible ways to do a report card, and is a modified Elem.FRM.
Statname.FRM	Lists the scores of one student along with the high, low, and mean for the class.
Whatif.FRM	This shows how you can write forms that calculate what students

need on future tests to achieve particular grades.

### **3. Commands**

This section describes the commands that can be used in **VGW**. To make it more understandable, they have been broken into command types:

- Non-printing commands: comments that are reminders for you, but don't print.
- Printer commands: fonts and other printer commands.
- Format commands: these are used to center lines, pages, and related things.
- Commands for writing data or switching data: the commands to actually print student data, or for selecting which data is printed.
- Repetitive commands: commands for writing lots of students or tasks.
- Statistical commands: commands that calculate and print statistical measures.

#### **(a) Non-printing commands**

Non-printing commands are words bracketed by "{" and "}" (see example 1, below). When the form is printed, these commands are removed from the form. The purpose of these, therefore, is to let you state what you have done in plain English so that the next time you look at the form, you will remember why you wrote it. These non-printing commands can be anywhere in the form. However, they cannot be nested (see example 2, below). If you want to put a left curly bracket into your text, put it in as two consecutive left brackets ("{{") (example 3). Note that if the right curly bracket is not present, the rest of the line is not printed (example 4).

Examples:

#	Statement in Form	Program Output
1	{This is a comment}	
2	{This is {a} comment}	comment}
3	{{This is a comment}	{This is a comment}
4	{This is a comment	

NOTE: For comments that take up an entire line, like example 4, that line will not be printed. This allows you to add a line at the top that explains the purpose of the form.

**(b) Printer commands**

Printing will use the setup you set on the Print Menu. However, there may be cases where you want to temporarily change the printer setup. Printer commands tell your printer to do special things. You can change the font, print in bold or underlined type, or change the font size.

These commands have the form: [Print“subcommand”] or [Print‘subcommand’], where subcommand is one of the subcommands listed below.

NOTE: All the commands listed below start and end with single or double quotes.

<u>SubCommand</u>	<u>Description</u>
BoldOn	Sets the printer to start printing with bold characters.
BoldOff	Turns off bold printing.
Chars=#	Sets the number of tenths of inches on a line, where “#” is a number. <i>Example:</i> [Print“Chars=60”] tells the program to set the width of a line to 5 inches.
Compressed	Sets the printer to print in compressed text (17.1 characters per inch, 10 lines per inch).
Courier	Sets the Courier font (non-proportional). If you have forms written for a non-proportional printer, they can be printed on a proportional printer without change if you use this command.
Elite	Sets the printer to print in elite text (12 characters per inch, 8 lines per inch).





<u>SubCommand</u>	<u>Description</u>
Font= <i>string</i>	Sets the font to <i>string</i> . The font can be any Windows font. The name to use for <i>string</i> is the one listed on the Font Dialog. <u>Example:</u> [Print“Font=Arial”] sets the font to Arial. If a font isn’t found, the command does nothing.
Footer=#	Sets the bottom margin to “#”, where # is in 1/10”. <u>Example:</u> [Print“Footer=10”] sets the bottom margin at 1”.
Header=#	Sets the top margin to “#”, where # is in 1/10”. <u>Example:</u> [Print“Header=10”] sets the top margin at 1”.
ItalicOff	Changes the font to no longer be italic.
ItalicOn	Changes the font to italic. The font must support italics.
Lines=#	Sets the number of tenths of inches on a page, where “#” is a number. <u>Example:</u> [Print“Lines=70”] sets the length of a page to 7 inches.
Normal	Sets the printer in the original font.
Offset=#	Sets the number of 1/10” spaces to be printed at the beginning of each line, where “#” is a number. The higher the number, the further from the left side of the page the document is printed. <u>Example:</u> [Print“Offset=5”] sets the left margin to one-half inch.
Offsetcm=#	This works the same as offset, but the value is in centimeters instead of inches.
Pica	Sets the printer to print in pica text (10 characters per inch, 6 lines per inch).
Points=#	For proportional printers, sets the font to that size type, in points. Pica = 12, elite = 10, and compressed = 8. You can set any value from 6 to 16 points.

Tab=#	Moves the printer in 1/10".
UOff	Sets the font to underlined.
ULOn	Turns off underlining.

**(c) Format commands**

These commands alter how the printed form will look.

<u>Command</u>	<u>Description</u>
CJust	Centers strings on a line. For a string to be centered, it must have a width specified. It complements the RJust and LJust commands.
CtrJust	Centers strings on a page. CJust centers within a width, CtrJust centers on the entire line of a page. See Attend.FRM or Progress.FRM to see the difference.
EndOfPage	Write a formfeed (end of page character). This starts a new page on the printer. Any forms longer than the number of lines on a page are automatically broken into pages. The EndOfPage command lets you pick where the pages will break if you want certain items to end up on the same page.
Exscore= <i>string</i>	Sets the word(s) to be printed when a student score is excused (below the minimum points). If [exscore], then it is set to “*Not submitted”, if [exscore=***], then "****" will be printed, if [exscore=], then nothing will be printed (also, see NoScore).
Just	Left justify strings. Names, dates, etc. can be written with a defined length (e.g., you can have the student names written with 25 characters). Left justification will write the names starting with the first character. <i>Example:</i> “Student Name                   ” (the quotes are there for illustration purposes, only). This is the standard way of writing names, etc.
NewLine	Causes a new line to be started.
NoLine	Prevents a new line (the next line will be written on the same line).

- NoScore=*string* Sets the word(s) to be printed when a student score is unentered. If [noscore], then it is set to “Not submitted”, if [noscore=ugh], then “ugh” will be printed, if [noscore=], then nothing will be printed (also, see ExScore).
- NotOnePerPage More than one student or set of students per page (default setting). See “OnePerPage”.

<u>Command</u>	<u>Description</u>
NoWay= <i>string</i>	For what-ifs, sets the word(s) to be printed when a student score is not possible. If [noway], then it is set to “Not possible”, if [noscore=oops], then “oops” will be printed, if [noscore=], then nothing will be printed.
OnePerPage	One student or set of students per page of output. Page breaks will be inserted after each student or set of students.
PageNum	Print page numbers on the pages.
RJust	Right justify strings. Names, dates, etc. can be written with a defined length (e.g., you can have the student names written with 25 characters). Right justification will write the names from the rightmost character. <i>Example:</i> “ Student Name” (the quotes are there for illustration purposes, only).
TwoPerPage	After two forms are printed, this starts a new page. This command works like OnePerPage, except starts a new page after every other student instead of after every student.
Zeroscore= <i>string</i>	Sets the word(s) to be printed when a student score is unentered and set to zero. If [zeroscore], then it is set to “0”, if [zeroscore=zero], then “zero” will be printed, if [zeroscore=], then nothing will be printed.

#### **(d) Commands to write or switch data**

There are quite a number of commands for accessing or switching data. These are listed below in several groups:

- Student data: prints data from an individual student.
- Task, attendance, and database data: data for tasks, attendance, and database items such as names, selecting the items, etc.
- Class or generic data: data relevant to the class, such as the class name, teacher, and section.

For most of the data, you can tell the program how to format the output. For example, almost all of the commands let you specify the number of characters to write the data. This is useful for formatting columns. *Example:* [Name:25] tells the

program to write the student name 25 characters wide. Since the name is less than 25 characters, the rest will be written with blanks. [Name] will write only the name, with no extra blanks.

For numeric data, you can tell the program how many numbers should be written after the zero. *Example:* [TaskScore:10:3] will write the data 10 characters wide, with 3 numbers after the zero: xxxxxx.xxx, where x is any number (notice that one space is taken by the period). If you specify a length that is less than the required spaces, the entire piece of data is written. *Example:* [TaskScore:0:3] will write the data with as many spaces as needed, but including three numbers after the decimal.

*NOTE: Formatting of commands, which are the “: #” seen after the commands listed below, are optional.*

## I. Class or Generic Data

These commands access class data, switch between classes, or access things like the current date.

<u>Command</u>	<u>Description</u>
“”:#	Anything between the double quotes (“”) is printed. This command is useful for specifying the width of headers, etc. when formatting tables.
AddSect:#	Adds # to the list of sections to use.
AttTotal:#	Prints the total number of class days.
Class= <i>string</i>	Switches the class to another one. The <i>string</i> must either specify a valid filename of a class, or specify nothing. If it specifies nothing, the original class will be used as the current class. Up to 7 classes in addition to the original one can be kept in memory at once (unless you run out of memory). If you have the same students in several classes or subjects, this can let you write one report form that will put all their data onto one printout. If <i>string</i> is a ?, VGW will ask you for the class name. If <i>string</i> is a ?#, where # is from 0 to 10, the program will either ask for the class or select that class number, if the class was already entered.
ClassName:#	Prints the name of the current class. <i>Note:</i> This

command does not print the file name, but instead the name you specify for printouts by selecting “Write Class name for printouts” from the Main Menu.

Currsect:#	Changes the section to use to the #. It removes any other sections from the list of sections to use.
<u>Command</u>	<u>Description</u>
Date:#	Prints the current date.
SectComm	Prints the section name or comment for the current section. Section comments are set in the program by selecting “Write Section names or comments”.
Teacher:#	Prints the teacher's name. The name must have been entered in the program by selecting “Write Teacher name”.

## II. Student Data

These commands access the student data, including grades and attendance.

<u>Command</u>	<u>Description</u>
AttAct:#	Prints the number of days the student has had an “activity”.
AttA2:#	Prints the number of days the student has had an “activity 2”.
AttAllA:#	Prints the total number of days the student has been absent. Absent includes excused, excused 2, unexcused, unexcused 2, other, and other 2.
AttAllP:#	Prints the total number of days the student has been present. Present includes present, present 2, late, late 2, activity, and activity 2.
AttChar:#	Prints a character that corresponds to the type of attendance the current student had on the current attendance day (AttDay). e.g., if the student was late, it prints: “L”.

AttExcused:#	Prints the number of days the student has had an “excused” absence.
AttE2:#	Prints the number of days the student has had an “excused 2” absence.
AttLate:#	Prints the number of days the student has been “late”.
AttL2:#	Prints the number of days the student has been “late 2”.

<u>Command</u>	<u>Description</u>
AttNotPres:#	Prints the number of days the student has been absent from the class.
AttOther:#	Prints the number of days the student has been “other” from the class.
AttO2:#	Prints the number of days the student has been “other 2” from the class.
AttPresent:#	Prints the number of days the student has been “present”.
AttP2:#	Prints the number of days the student has been “present 2” in the class.
AttString:#	Prints the name of the attendance type for the current student on the current attendance day. e.g., if the student was late, it prints: “Late”.
AttUnex:#	Prints the total number of “unexcused” absences by the student.
AttU2:#	Prints the total number of “unexcused 2” absences by the student.
Comment	Prints automatic comments. The comment is picked from the student score of the current task. For comments to be printed, the automatic comments must have been entered (which are entered by selecting “Write Reports”, then “Add automatic comments”).
Comment?	Asks for the comment, then prints it. There are several choices. You can pick an automatic comment, you can print blank lines, underlined lines, or you can type in lines on the keyboard. A menu appears that lets you choose which.
DBVal:#	Prints the database data of the current student for the current database item.
Firstname:#	Prints the first name of the current student. A first



name is either the first word in the student name, or everything past the comma.

FName:# Prints a flipped name--where the first and last names are reversed.

<u>Command</u>	<u>Description</u>
ID:#	Prints the ID of the current student.

LastName:# Prints the last name of the current student. A last name everything before a comma, or the last word in the student name.

Name:# Prints the full name of the current student.

Section:# Prints the section of the current student.

TaskPer:#:# Prints the percentage score that the current student scored on the current task.

TaskPts:#:# Prints the total attempted points of the student on the current task. If all the task scores have been entered, it will be the same as TaskTotal, below.

TaskScore:#:# Prints the score on the current task for the student.

Note: the following are analogous to the WhatIf part of the program.

<u>Command</u>	<u>Description</u>
WhatIf“xx”	Set the grade xx, where xx is any valid grade. Example: WhatIf“A” sets the current WhatIf grade to an A.

WhatIf“ptsleft”:#:# Points left to achieve the grade set by WhatIf“xx”.

WhatIf“ptsneeded”:#:# Points needed to achieve the grade set by the WhatIf“xx” command.

WhatIf“%needed”:#:# Percent of the points needed to achieve the grade set by the WhatIf“xx” command. It is equal to  $100 * (\text{ptsneeded} / \text{ptsleft})$ .

WhatIf This prints all the grades and the needed

points and percents, just like in the Whatif section of the program.

### III. Task, attendance and database data

These commands are used to change or specify the tasks, attendance, or database items to use. In general, you want to change or specify the current task with Task or DTask, then print the data with the other commands.

<u>Command</u>	<u>Description</u>
AttDay#	Changes the current attendance day to #.
AttName:#	Prints the name of the current attendance day.
AttNumber:#	Prints the number of the current attendance day.
AttTotal:#	Prints the total number of class days.
Currsect:#	For seating charts, sets the current section to '#'. It also removes any other sections from the list of used sections.
DBName:#	Prints the name of the current database item.
DBNum:#	Prints the number of the current database item.
DB“string” or DB ‘string’	If a name is between the quotes, the first database item that includes that name is made the current database item. To make sure you have the right item, it is usually best to put the entire item name between the quotes.
DB#	Makes the # the current database item.
DTask	Defines a task as the current one. No data is printed. The task can be specified many ways: [DTask#x], where $x$ is the number of the task. [DTask@x], where $x$ is the $x$ th task you have defined. [DTask(x)], where $(x)$ is the total number of tasks in the class / $x$ . <u>Example:</u> If there are ten tasks defined,

[DTask(2)] is the fifth task.

[DTask?]. “?” means that the program will ask you to select the task when it reaches that point in the report form.

[DTask+]. The “+” can be added to any of the first three definitions, and tells the program to increment the number by one.

[DTask+] will increment the current task by one. [DTask(2)+] will, if there are 16 tasks, make the current task the ninth one ( $16/2 = 8$ , then add one).

[DTask"string"], where *string* is any word or characters, will select the first task that matches the characters. So [Dtask “First”] will select any task with first in the name (e.g., “First Exam” or “The first test”).

<u>Command</u>	<u>Description</u>
DTask (cont)	In some circumstances, like the [Loop] command described below, just a plain [DTask], without any designation of a number, is required. When [Loop] is used, the plain [DTask] is modified to have the required number for the loop.

Task:##

This works like DTask, except that, after the current task has been changed, the score for the current student on that task is printed.

TaskDef

Prints the definition of the current task.

TaskName:#

Prints the name of the current task.

TaskNum:#

Prints the number of the current task.

TaskTotal:##

Prints the total points of the current task.

### (e) Repetitive commands

These are commands that are used to write lists of students or tasks. There are three commands that repeat through a series of items: repeat, loop, and students. There is also a related command, header, that prints a header at the top of a page. Last, there are items that filter values, so that only ones that meet a condition are printed.

<u>Command</u>	<u>Description</u>
AttRepeat:##	Repeats the attendance days from the first # to the second #. If the

#s are not included, the first is set to 1, the second one is set to the last attendance day. It can also be used with the [IFAtt] commands to print only certain types of attendance.

#### DFilter

This command sets a value to test for a database item. The values needed to be passed can be above (“>”) a value, below (“<”) a value, between two scores (“^”), or equal to a value (“=”). A “?” will make the program ask for the value. If equal is chosen, any matches to the value qualify as equal. If between is chosen, the two values are separated by a comma. When testing, lowercase values are converted to uppercase ones.

*Examples:*

[DFilter?] makes the program ask for the filter to use.

[DFilter<K] passes values below “K”.

[DFilter=the] passes any db value that has “the” in it.

[DFilter^C,T] passes any values from D to S.

#### Command

#### Description

##### Header

For the following repetitive commands, [Header] will print whatever is on the header line at the beginning of any new page. This command will let you write long lists of students or task scores, each with a header on the top of any new pages.

#### IfAttx

These commands are used with AttRepeat. If the current student has that attendance, the line is printed, otherwise, the attendance day is incremented. x can be:

A (activity)	A2 (activity 2)
E (excused)	E2 (excused 2)
L (late)	L2 (late 2)
O (other)	O2 (other 2)
P (present)	P2 (present 2)
U (unexcused)	U2 (unexcused 2)
N (not found)	

*Example:* [AttRepeat][IFAttU] [AttName] will print out the dates for all unexcused absences for the current student.

*Example:* [AttRepeat][IFAttL2] [AttName] will

print out the dates for all late 2 days for the current student.

- IfStudentExists Prints what is left on the line only if the student exists. It's use only makes sense when the "Class=" command is used. If the student doesn't exist, a line feed is not done.
- IfDFilter If the student does not pass the filter value for the current database item, the rest of the current line is skipped. The filter value is set with "DFilter".
- IfTaskExists Prints what is left on the line only if the current task exists and the current student's score on the task is at least the minimum allowed. If the task doesn't exist, or the score is not entered, a line feed is not done.
- IfTaskMiss Prints what is left on the line only if the current task score is unentered (missing). This is the opposite of IfTaskExists.
- IfTFilter If the student does not pass the filter value for the current task, the rest of the current line is skipped. The filter value is set with "TFilter".

## Command

### Loop

## Description

Repeat more than one line. Any lines between [Loop] and [Loopend] will be repeated. Note that the lines containing [Loop] and [Loopend] are not printed. Tasks specified with the commands [Task] and [DTask] that are found between the [Loop] and [Loopend] commands are incremented between repetitions.

*Note:* [DTask#I], [Task@I], etc, will not be incremented within Loop, so use plain [DTask] and [Task]. The [Loop] command can be modified by telling the program how to increment the Tasks.

[Loop:#:#] is used specify, in order the beginning task to use and how much to increase the task numbers each loop.

*Example:* [Loop:3:2] will print tasks, starting with the third task and increases the task numbers by 2 each time the loop is repeated. If the “#” numbers are omitted, it is assumed that the loop should start at the first task and increment them by one each time.

[Loopend] can also be modified. In [Loopend:#], the “#” is the number of the last task to be printed.

*Example:* [Loop:3:2] [Task] [Loopend:7]

This example will print out the scores for the third, fifth, and seventh tasks. The loop will end when the last task in the class is reached. So, in the above example, if there were only five tasks in the class, only scores for the third and fifth tasks will be printed.

### Repeat#

Write the information on the line, but increment the task numbers on that line by one each time the line is repeated. “#” is a number.

If “#” is not present, the line will be repeated once for each task in the class.

If “#” is a number, the line is repeated that many times.

If “#” is a number in parentheses, like “(2)”, the line will be repeated by the quantity: the total number of tasks in the class / that number.

*Examples:* If a class has 16 tasks that have been

defined,

[Repeat] will repeat the line 16 times;  
[Repeat(2)] will repeat the line 8 times;  
[Repeat5] will repeat the line 5 times.

Note that, in each case, any task numbers on that line will be incremented by one each time the line is repeated.

<u>Command</u>	<u>Description</u>
Repeat (cont)	If “#” is a question mark, when the form is printed you are asked for a list of the tasks to write. The line will be repeated enough times to write out that list.
Students	Write the information on the line for each student in the class, from the first student to the last one. The purpose of this command is to easily write out a list of students' data. <i>Example:</i> [Students][Name] will write the names of all the students in the class, one per line, starting with the first student and ending with the last student.
TFilter	This command sets a value to test for a task score. The values needed to be passed can be above (“>”) a value, below (“<”) a value, between two scores (“^”), or equal to a value (“=”). A “?” will make the program ask for the value. If “between” is chosen, the two values are separated by a comma. All values must be numbers. If you want to test letter grades, use the grade values for the testing.  <i>Examples:</i> [TFilter?] makes the program ask for the filter to use.  [TFilter>90] passes values above 90. [DFilter=100] passes any score that is 100. [DFilter^70,90] passes any values above 70 and below 90.

**(f) Statistical commands**

These are commands that calculate and write various statistical values, like averages, means, medians, and standard deviations. The “Avg” commands, below, are used to print averages of student scores listed on the printout. The form RCElem.FRM gives an example of their use. The other statistical measures are seen in Statname.FRM and Stats.FRM.

<u>Command</u>	<u>Description</u>
AvgPct:##	Prints the percentage for the student scores between AvgStart and AvgStop.
AvgResume	Resumes averaging without restarting. With AvgStop, this command allows you to skip some student scores.



AvgShow:##	Prints the average of all the student scores between AvgStart and AvgStop.
AvgStart	Starts averaging task scores.
AvgStop	Stops averaging task scores.
<u>Command</u>	<u>Description</u>
AvgSum:##	Prints the sum of the student scores between AvgStart and AvgStop.
Highscore:##	Prints the high score for the current task.
Lowscore:##	Prints the low score for the current task.
Mean:##	Prints the mean for the current task.
Median:##	Prints the median for the current task.
StdDev:##	Prints the standard deviation for the current task.

#### 4. Designing a form

To design a form, you should first read Sections 1 through 3. Second, you should look at and study the report forms that are included with the program. These are detailed below in section 5. Then:

- Use a word processor or the editor in **VGW** to write what you want the form to look like. Just type in what you want.
- Put square brackets around those parts that are changeable and need to be put in by the program--the data you previously entered into the program.
- Replace what you have in the square brackets with commands.
- Put in format commands (widths, decimal places, etc).
- Write the form to disk as a plain text (ASCII) file.
- Test the form from the program. Make a note of the problems and errors, then repeat steps through until it looks how you want. It may take a couple of iterations to make it look how you want.

**Example**

If you want to design a form to print the second grade for all students in a class. How do you do it?

- Use a word processor or text editor to write the following:

```
Class: Test class
Date:  xx/xx/xx
```

```
Name                Task #2
first name           100.0
second name          90.0
```

- Next, notice that there are five types of data present: the class name, the date, the task name, the student names, and the task scores.
- Replace the fake data in part with brackets:

```
Class: [Test class]
Date:  [xx/xx/xx]
```

```
Name                [Task #2]
[first name]         [100.0]
[second name]        [90.0]
```

- Now, look up the commands to replace what is inside the brackets. For the student names, you want a repetitive command. For the task scores, you should also specify the digits and decimal places.

```
Class: [Classname]
Date:  [date]
```

```
Name                [dtask#2] [taskname]
[students] [name]           [taskscore:5:1]
```

- Next, specify formatting for the student list. To make sure things line up, you can specify a width for “Name”, as well.

```
Class: [Classname]
Date:  [date]
```

```
[“Name”:20] [dtask#2] [taskname:25]
[students] [name:20] [taskscore:25:1]
```

- Last, save the form as plain text and test it in the program. If changes are needed, repeat these steps.

- If you replace [dtask#2] with [dtask?], when you run the form, **VGW** will ask you to select the task.

## XII. OTHER PROGRAM FEATURES

### A. Class Maintenance

**VAR Grade for Windows** has a complete file maintenance section. You can copy, move, delete, recover, or rename any or all of your class files. Class files end in “.PAR” and “.DAT”. When you select or change classes, the “.PAR” ending is implied, and the “.DAT” file is also changed.

#### 1. Renaming class files

To change the name of a class file, there are two methods:

- Select “File Rename”. Next, select the file to rename. Last, enter the new name.
- Select “File Save as”. Next, select a new name. If you don’t want a copy of the class with the old name, select “File Delete”, and select the old class file name.

#### 2. Copying class files

Copying a class file is frequently done to save a copy of the files on a floppy disk.

- Select “File Copy”. Next, select the file to copy. Last, select a new drive, directory, and/or filename.

#### 3. Moving class files

Moving class files will copy the specified classes into the new directory or drive, then delete them from the original place. This option is useful for archiving classes after the semester or year is finished.

- Select “File Move”. Next, select the file to move. Last, select a new drive and/or directory.

#### 4. Deleting class files

This option will delete all the specified class files. *WARNING: This option will remove whatever class files you specify. Be sure you have picked the ones you really want to delete.* Note that you are asked if you are sure that you want to delete those classes

before the program actually removes them.

- Select “File Delete”. Select the files to delete. When asked whether you really want to delete the files, select “Yes”.

## 5. Recovering backup classes

Backup files are normally made of your classes. These backups are copies of the class from the previous time you altered it. The files end in the extensions “.PAX” and “.DAX”. If you have backup class files, and accidentally delete or damage your class, you can use the File Recover command to restore your class. Note that the class is restored from the previous time the data was saved.

- Select “File Recover”. Next, select the backup to recover. The class is read into the program, and is now a regular class.

## 6. Write/Not write backup files

Backup files are normally made of your grades. These backups contain the data from the *previous* time that you saved your data, and are this program's equivalent to the BAK files your word processor probably produces. The backup files have the same name as the class, but have the extensions “.DAX” and “.PAX”.

No matter which type of file the class data is written (read-only, hidden, or normal), the backup files are made normal files. This means that they can be deleted should you not want to clutter up your disk with them. However, if you prefer not to have the program make backup files, you can use this option to prevent backups.

Select “Option File protection”. Uncheck the option “Make backups when saving data”.

It is *strongly* recommended that you let back-up files be made and that you let the program write the class files as read-only. The combination of these two options will give your data much greater security.

## 7. Change class file protection

The class grading files are, by default, written in what is known as “read-only” files. This means that if you are in DOS or the Windows File Manager, and ask that the files be deleted, they will *not* be deleted. The program does this to prevent accidental deletions of the data, but this can sometimes cause problems with other programs, most notably some

backup programs. You can change the way files are written by specifying that they be written normally (they can be deleted by DOS), as read-only files, or that they be “hidden” files (DOS will not even list the files on a “dir” command). Hidden files will still be seen by **VAR Grade for Windows**, but not by anyone casually snooping around your disk.

## 8. Preventing Data Loss

It is very important to prevent data from being lost. There are several things that the program does to prevent this loss.

- You cannot exit the program without telling it whether you want your data saved.
- Files are written in read-only format by default. You cannot delete or erase these files in DOS. The default can be changed to “normal” (if you do not fear deleting your class grades), or “hidden”. This is explained in Section 7.
- When files are written to disk, the previous class data is turned into backup files. The file CLASSNAME.PAR is changed to CLASSNAME.PAX. The file CLASSNAME.DAT is changed to CLASSNAME.DAX. These backup files are not read-only, and can be deleted by DOS. Should you want to delete old class files, use the DOS 3.x (or later) command `Attrib` to change the file attribute to “normal”, then delete the file.

*Example*, type:

```
A> ATTRIB -R CLASSNAME.*A> DEL CLASSNAME.*
```

You can tell the program to stop making backup files as described in Section 6, but we recommend that you do this only if disk space is at a premium.

There are also some things **you** can do to prevent data loss:

- Save your class data to a floppy. Having two or more copies is safer than one copy.
- Print your data after entering them. Select “Write by Section or class”, then one of the options to save the data you just entered.
- Periodically check your computer for viruses. There are shareware and non-shareware products that will detect most viruses.





## B. International Support

There are several features in **VAR Grade for Windows** that make it easier for users outside the United States. Some of these have been mentioned previously, like defining your own grading system. Two more are discussed here.

### 1. Dates

The date format is taken directly from Windows. However, you can change it. The names for the days of the months and days of the week can also be changed. Chapter VIII, Section A gives explicit details on how to change either of these.

### 2. Output terms

You can change *any* words (“strings”) that are printed. Some of these strings also affect the screens in the program. For example, changing the names of the month names, described above, changes the names on the calendars. The main purpose of being able to change these words is to let non-English speakers translate the output strings into their own language. However, if you dislike the titles or names of some or all of the output, you can change them!

Each screen that prints data has an “Output terms” dialog or two. However, all of them are collected in “Option Output terms” on the Main Menu. The types of strings you can change are broken into several categories. Many of the strings affect most or all output (e.g., “Name”, “ID”, etc.), while others are specific to particular parts of the program. Hence, it is hard to put them into categories based upon this report or that graph. Instead, they are put into general categories. There are several general categories:

- Common terms: terms found in most printouts, such as “Class” or “Date”.
- Task names: names of each of the types of tasks.
- Yes no: the character used for Yes and No input.
- Task definition terms: a variety of terms used to describe tasks, such as “Task type” and “Cutoffs”.
- Attendance and task terms: items used in printing attendance or task scores, such as “Attendance”, “Day”, “Assignment”, or “Missed days”.
- Statistics terms: terms used to describe statistics, such as “High score”, “Mean”,

or “Total points”.

- Statistics description terms: terms found in the Descriptive Statistics part of the program.
- What if terms: terms found in the What if part of the program.
- Assign grades terms: terms found in the Assign grades part of the program.

After you have changed any of the strings, they are written to a file called VGW.INT. Every time **VAR Grade** starts, it searches for the file. If it finds it, it uses those strings in preference to the default ones. You can return to the defaults by deleting or renaming VGW.INT.

If the file VGW.INT is not found, it searches for a file called VGW.xxx, where xxx is the international phone code for your country. For example, to phone the US, you would dial 001, then the US area code and phone number. If you are using the program in the US, **VAR Grade for Windows** will search for VGW.INT, if it doesn't find that, it will search for VGW.001. If it doesn't find that, it uses the program defaults. In the future, we plan to send out translations for non-English languages as VGW.xxx files. You can make your own VGW.xxx file by translating the strings, then renaming VGW.INT to the appropriate VGW.xxx file.

### 3. BWCC files

One file sent with VAR Grade for Windows is titled BWCC.DLL. This file is used to make dialogs and windows, and it includes the buttons such as “OK” and “Cancel”. This file is in English. However, versions exist for other languages, which will have the terms in that language. You can find BWCC.DLL on bulletin boards throughout the world, often in sections devoted to Borland International products.

## C. Program Parameters--Defaults and Limits

The default values are those that are there when the program starts.

<u>Parameter</u>	<u>Default</u>	<u>Maximum</u>
Students	0	16000
Sections	1	16000
Tasks (total)	0	8000
Total points for a number task	100	1000000
Total points per sum task	-	9999999

Attendance Days	0	240
Database items	0	200

## **D. Beeps or Noises**

If the computer beeps or makes a noise you probably entered data that was not in the correct range or in the allowed set of characters. If you select “**Edit Options**”, then uncheck “**Make sound on data input errors**”, data entry beeps will no longer occur.

# GLOSSARY

Archive	Files that have been compressed so that they take up less space on the disk. They need to be unarchived to be used.
ASCII	A standard code that translates normal characters and symbols into 7 bit binary computer code. ASCII files are in ASCII characters, and are also called plain or normal text.
Cell	A highlighted area of the screen that holds one piece of data.
Comma & Quote Files	Files that have strings of characters between double quotes (“”), and commas separating pieces of data. Numbers are separated only by commas.
Cursor Keys	Keys that move the position of the cursor on the screen. They include the arrow keys, the home, end, page up, and page down keys.
Database	Data kept by a program in a structured way. The data can then be sorted, abstracted, graphed, or otherwise manipulated.
Default	The value when the program starts.
Directory	Group of files on a disk. Files on hard disks are usually grouped together in directories to make it easier to find the file you need. Drives are also directories.
Fixed Disk	A disk that cannot be removed from the computer.
Flat Files	Files that have data in columns, separated by spaces.
Floppy Disk	A disk that can be removed from a drive.
Hard Disk	A disk that cannot be removed from the computer.
Install	Put the files needed by a program onto new

disks such that the new disks have the program files in the places needed by that program.

#### Main Menu

The menu on the main window that lists the major sections of the program. This is the menu seen when you first enter the program.

Path	A list of directories that tells MS-DOS where executable programs are. The path is set by typing “SET PATH=directory1;directory2;” at the DOS prompt, where directory1 and directory2, etc., are names of directories.
RAM	Random access memory. This is memory the computer uses to run programs. Computers that run Windows have 2 megabytes or more of RAM, and <b>VAR Grade for Windows</b> needs 4 megabytes of RAM to run. Memory on floppy or hard disks is not RAM.
Regular tasks	Those tasks that you can put in new data. They are the number, letter, and extra credit tasks.
Special keys	Special keys are those that do not print normal (ASCII) keys. The function keys (<F1> through <F10>), the <Alt>, <Ctrl>, <Ins>, <Del>, and the cursor keys.
Special tasks	Those tasks that are calculated by the program. You cannot put in new scores into these tasks.
Strings	A series of words, such as names, addresses, or any other words.
Task	A graded assignment for the class, or a calculated value based on assignments.
Toggle	Turn on and off.
Wildcard	The characters * and ? are used by MS-DOS to match more than one character, as a wild card, when playing cards, can match any other card.