Overview

The **HP-12C** calculator is a **programmable** financial calculator utilizing **reverse polish notation** with a **4 register stack**, **99 program lines** available for **programs**, **financial registers**, **20 data storage registers**, and **continous memory**.

Functions include **percentage**, **statistics**, **linear estimation**, **calendar**, **financial**, **depreciation**, **bond price and yield**, **discounted cash flow analysis**.

Calculator programs feature **conditional branching** and **looping**. Also, the calculator has the ability to store **multiple programs**.

Applications for this calculator include the fields of **real estate** and **lending**, **mortgage brokering**, **stocks** and **bond trading**, **leasing**, and **banking**.

Keyboard Functions

ON Turns the calculator off and exits the program, retaining continous memory.

f Press to access gold functions.

g Press to access blue functions.

PREFIX Displays mantissa of number in the displayed X-register.

ENTER Pushes number in display into the stack.

CHS Changes sign of number or exponent of 10 in displayed X-register.

EEX Enter exponent. After pressing, next numbers keyed in are exponents of 10.

0-9 Digits used for keying in numbers and display formatting. To format the display, press **f** then a digit representing the number of decimal places desired in the display.

. Decimal point. Also used for display formatting. Pressing **f .** sets the calculator to display numbers in scientific notation.

CLX Clears contents of displayed X-register to zero.

+,-,x,/ Arithmetic operators. Computes Y-register operator X-register and places the result in X-register after the stack is dropped.

STO Store. Followed by number key, decimal point and number key, or top-row financial key, stores displayed number in storage register specified. Also used to perform storage register arithmetifc.

RCL Recall. Followed by number key, decimal point and number key, or top-row financial key, recalls value from storage register specified into the displayed X-register.

REG Clears contents of stack (X,Y,Z,T), all storage registers, statistical registers, and financial registers. Leaves program memory untouched; not programmable.

% Computes x% of y and retains the y value in the Y-register.

DLT % Computes percent of change between number in Y-register and number in displayed X-register.

%T Computes percent that x is of number in Y-register.

D.MY Sets date format to day-month-year; not programmable. (Example: 2.121995 is December 2, 1995) When this mode is set, the **D.MY** status indicator in the display is lit.

M.DY Sets date format to month-day-year; not programmable. (Example: 2.121995 is February 12, 1995)

DATE Changes a date in the Y-register by the number of days in the X-register and displays day of week. Sunday is 7. Monday is 1.

DYS Computes the number of days between two dates in the Y- and X-registers. The answer shown in the display is the actual number of days between the two dates, including leap days (the extra days occuring in leap years), if any. In addition, the HP-12C also calculates the

number of days between the two dates on the basis of a 30-day month. This answer is held inside the calculator; to display it, press **x><y**.

FIN Clears contents of financial registers.

BEG Sets payment mode to Begin for compound interest calculations involving payments. When this mode is set, the **BEGIN** status indicator in the display is lit.

END Sets payment mode to End for compound interest calculations involving payments.

INT Calculates simple interest. Key in number of days, then press **n**. Key in annual interest rate, then press **i**. Key in the principal amount, then press **CHS PV**. Negative, due to the cash flow sign convention. Press **f INT** to calculate and display the interest accrued on a 360-day basis. If you want to display the interest accrued on a 365-day basis, press **R x><y**. Press **+** to calculate the total of the principal and the accrued interest now in the display. The quantities **n**, **i**, and **PV** can be entered in any order.

n Stores or computes number of periods in financial problem.

12x Multiplies a number in displayed X-register by 12 and stores the resulting value in **n**.

i Stores or computes interest rate per compounding period.

12/ Divides number in displayed X-register by 12 and stores the resulting value in **i**.

PV Stores or computes present value (initial cash flow) of a financial problem.

PMT Stores or computes payment amount.

FV Stores or computes future value (final cash value) of a financial problem.

AMORT Amortizes x number of periods using values stored in **PMT**, **i**, **PV**, and the display. Updates values in **PV** and **n**. Press **f CLEAR FIN** to clear the financial registers. Enter the periodic interest rate, using **i** or **12/**. Enter the amount of the loan (the principal), using **PV**. Key in the periodic payment, them press **CHS PMT**. (The sign of **PMT** must be negative, in accordance with the cash flow sign convention.) Press **g BEG** or (for most direct reduction loans) **g END** to set the payment mode. Key in the number of payments to be amortized. Press **f AMORT** to display the amount from those payments applied toward interest. Press **x><y** to display the amount from those payments applied toward the principal. To display the number of payments just amortized, press **R R**. To display the remaining balance of the loan, press **RCL PV**. To display the total number of payments amortized, press **RCL n**.

NPV Calculates the net present value of up to 20 uneven cash flows and initial investment using values stored with **CFo**, **CFj**, and **Nj**. Press **f CLEAR REG** to clear the financial and storage registers. Key in the amount of the initial investment, press **CHS** if that cash flow is negative, then press **g CFo**. If there is no initial investment, press **0 g CFo**. Key in the amount of the next cash flow, press **CHS** if the cash flow is negative, then press **g CFj**. If the cash flow amount is zero in the next period, press **0 g CFj**. Repeat last step for each cash flow until all have been entered. Enter the interest rate, using **i** or **12/**. Press **f NPV**.

IRR Calculates the internal rate of return (yield) for up to 20 uneven cash flows and initial investment using values stored with **CFo**, **CFj**, **Nj**. Enter the cash flows using the method described for **NPV**. Press **f IRR**.

The internal rate of return is shown in the display and also stored in the **i** register. If you get an **Error 3** message in the display, you can try the procedure outlined in Error Conditions regarding an **Error 3** message to resolve this condition by entering a new guess for **IRR**.

CFo Initial cash flow. Stores contents of displayed X-register in R0, initializes **n** to zero, sets N0 to 1. Used at the beginning of a discounted cash flow problem.

CFj Cash flow j. Stores contents of X-register in Rj, increments **n** by 1, sets **Nj** to 1. Used for all cash flows except the initial cash flow in a discounted cash flow problem.

Nj Stores the number (from 1 to 99) of times each cash flow occurs as **Nj**. Assumes 1 unless otherwise specified.

PRICE Calculates bond price, given desired yield to maturity. Enter the desired yield (as a percentage) using **i**. Enter the annual coupon rate (as a percentage), using **PMT**. Key in settlement (purchase) date, then press **ENTER**. Key in the maturity (redemption) date. Press **f PRICE**. The price is shown in the display and also is stored in the **PV** register. The interest accrued since the last interest date is held inside the calculator: to display the interest, press **x><y**; to add the interest to the price, press **+**.

YTM Calculates yield to maturity, given bond price. Enter the quoted price (as a percent of par: HP-12C uses a par value of 100 for bond calculations), using **PV**. Enter the annual coupon rate (as a percentage), using **PMT**. Key in the settlement (purchase) date, then press **ENTER**. Key in the maturity (redemption) date. Press **f YTM**.

SL Calculates depreciation using straight-line method. For depreciation calculations, enter the original cost of asset, using **PV**. Enter the salvage value of asset, using **FV**. Enter the expected useful life of the asset (in years), using **n**. **SL**, **SOYD**, and **DB** each place the amount of depreciation in the display. To display the remaining depreciable value (the book value less the salvage value) after the depreciation has been calculated, press **x><y**.

SOYD Calculates depreciation using sum-of-the-years digits method. Enter cost of asset, salvage value of asset, and expected useful life of asset, as described for **SL**.

DB Calculates depreciation using declining-balance method. Enter cost of asset , salvage value of asset, and expected useful life of asset, as described for **SL**. Enter the declining-balance factor (as a percentage), using **i**. For example 1.25 times the straight-line rate-- 125 percent declining-balance-- would be entered as **125 i**.

SUM Clears statistical storage registers R1 through R6 and stack registers.

SUM+ Accumulates statistics using numbers from X- and Y-registers in storage registers R1 through R6. Key the y-value into the display. Press **ENTER**. Key the x-value into the display. Press **SUM+**. Each time you press **SUM+**, the calculator does the following: The number in R1 is increased by 1, and the result is copied into the display. The x-value is added to the number in R2. The square of the x-value is added to the number in R3. The y-value is added to the number in R4. The square of the y-value is added to the number in R5. The product of the x- and y-values is added to the number in R6.

SUM- If you discover you have entered data incorrectly, the accumulated statistics can easily be corrected: If the incorrect data point or data pair has just been entered and **SUM+** has been pressed, press **g LST x g SUM-**. If the incorrect data point or data pair is not the most recent one entered, key in the incorrect data point or data pair again as if it were new, but press **g SUM-** instead of **SUM+**. These operations cancel the effect of the incorrect data point or data pair. You can then enter the data correctly, using **SUM+**, just as if it were new.

x bar Computes mean (average) of x-values and y-values using accumulated statistics. The mean of the x-values appears in the display after the function key is pressed; to display the mean of the y-values, press **x><y**.

x bar w Computes weighted average of y-(item) and x-(weight) values using accumulated statistics.

s Computes sample standard deviations of x- and y-values using accumulated statistics. The standard deviation of x-values appears in the display after **s** is pressed; to display the standard deviation of the y-values, press **x><y**.

y hat,r Linear estimate (X-register), correlation coefficient (Y-register). Fits a line to a set of (x,y) data pairs enterd using **SUM+**, then extrapolates this line to estimate a y-value (y hat) for a given x-value. Also computes strength of linear relationship (r) among that set of (x,y) data pairs.

x hat,r Linear estimate (X-register), correlation coefficient (Y-register). Fits a line to a set of (x,y) data pairs enterd using **SUM+**, then extrapolates this line to estimate a x-value (x hat) for a given y-value. Also computes strength of linear relationship (r) among that set of (x,y) data pairs.

SQRT Computes square root of number in displayed X-register.

y^x Raises number in Y-register to power of number in X-register.

1/x Computes reciprocal of number in displayed X-register.

n! Computes factorial [n*(n-1)...3*2*1] of number in displayed X-register.

e^x Natural antilogarithm. Raises e (approximately 2.718281828) to power of number in displayed X-register.

LN Computes natural logarithm (base e) of number in displayed X-register.

RND Rounds mantissa of 10 digit number in X-register to match the display.

INTG Leaves only the integer portion of number in displayed X-register by truncating fractional portion.

FRAC Leaves only the fractional portion of number in displayed X-register by truncating integer portion.

x><y Exchanges contents of X- and Y-registers of stack.

R Rolls down contents of stack for viewing in displayed X-register.

LST x Recalls number displayed before the previous operation back into the displayed X-register.

P/R Program/Run. Toggles into and out of program mode. Automatically sets program to line 00 when returning to Run mode. When in program mode, the **PRGM** status indicator in the display is lit.

MEM Memory map. Describes the current allocation of memory; the number of lines allotted to program memory and the number of available data registers.

R/S Run/Stop. Begins execution of a stored program. Stops execution if program is running. If multiple programs are held in program memory, you can execute a specific program by positioning the calculator to the program line where that program starts by using the **GTO** function. Then pressing **R/S** will begin execution at that program line. When executed as a program, usually to display intermediate results, pressing **R/S** will resume execution of the program at the line just after the **R/S** instruction.

SST Single step. Displays line number and contents of next program memory line. If held down, in PRGM mode, displays line number and contents of all program memory lines, one at a time. In RUN mode, executes instruction, displays result, and moves to next line when released.

GTO Go to. Followed by a two digit number, in RUN mode positions calculator to that line in program memory. No instructions are executed. In PRGM mode, pressing **g GTO .** followed by two digit keys sets the calculator to the program line specified. As an instruction in a program, (omitting the **.** enters the keystroke sequence as a program instruction) causes calculator to branch to the specified line number next, and resumes program execution from there. The decimal point is not necessary if the calculator is in RUN mode, but it is necessary if the calculator is in PRGM mode.

PSE Pause. Stops program execution for about 1 second and displays contents of X-register, then resumes program execution.

BST Back Step. Displays, in PRGM mode, line number and contents of previous program memory line. In RUN mode, displays line number of previous program memory line when pressed; displays original contents of X-register when released. No instructions are executed.

x<=y
x=0 Conditional. **x<=y** tests number in X-register against that in Y-register. **x=0** tests number in X-register against zero. If true, calculator continues execution at next program memory line. If false, calculator skips next line before resuming execution.

Error Conditions

Some calculator operations cannot be performed under certain conditions (for example, divide when X-register = 0). If you attempt such an operation under these conditions, the calculator will display the word **Error** followed by a digit, **0** through **8**. Listed below are the definitions of each error number.

0 Mathematics error. Attempt to divide by zero. Also operations which would result in complex numbers (example: SQRT(-1)).

1 Storage Register Overflow. When performing storage register arithmetic the magnitude of a result is greater than 9.999999999E+99.

2 Statistics. Trying to perform a statistics function or linear estimation when improper condition exists.

3 **IRR**. The calculation of **IRR** is very complex. This error indicates that the given initial guess was not sufficiently close to the correct value. Make a guess for the interest rate and key it in. Press **RCL g R/S** to enter the new guess into the calculator. Then press **f IRR**. This guess value is not held in continous memory, and is only valid during the current session.

4 Memory error. Attempting to enter more than 99 program lines.

5 Compound interest and depreciation errors.

6 Storage registers and discounted cash flow register errors.

7 **IRR**. No answer exists for the given cash flows.

8 Calendar function errors. Usually date format is incorrect. When in D.MY mode the date must be entered via the following format: dd.mmyyyy and when in M.DY mode: mm.ddyyyy.

Pr Error Continous memory has been reset. Displayed when the program is started and cannot find an initialization file, usually placed in C:\WINDOWS. This is a way to reset Continous Memory: simply delete the initialization file before starting the program. All registers are set to zero and program memory is filled with the **g GTO 00** instruction.

At any time a number is too large to be displayed in the display, the display shows **9.999999 99**. This means that the last operation resulted in an overflow condition. If that should happen when a program is running the program will halt. Also when an operation results in a number less than 1E-99 then that number is replaced by zero.

Registering

To register send check or money order to:

TELEMARK SOFTWARE
4441-2 Via Sepulveda
San Diego, CA 92122

The registration fee is **$34.95** US. Foreign country orders add $5.00 US for shipping. CA residents add 7.25% Sales Tax. Please allow 10 days for delivery via first class mail.

The registered version of this program has all keyboard functions enabled and includes a copy of "**The HP-12C Pocket Guide: Just In Case**" by Grapevine Publications.

Other inquiries can be mailed to **shin@saturn.scripps.edu**.

Financial Calculations

The Financial Registers

In addition to the data storage registers discussed, the HP-12C has five special registers in which numbers are stored for financial calculations. These registers are designated **n**, **i**, **PV**, **PMT**, and **FV**. The first five keys on the top row of the calculator are used to store a number from the display into the corresponding register, to calculate the corresponding financial value and store the result into the corresponding register, or to display the number stored in the corresponding register.

Storing Numbers Into the Financial Registers

To store a number into a financial register, key the number into the display, then press the corresponding key (**n**, **i**, **PV**, **PMT**, or **FV**).

Displaying Numbers in the Financial Registers

To display a number stored in a financial register, press **RCL** followed by the corresponding key.

Clearing the Financial Registers

Every financial function uses numbers stored in several of the financial registers. Before beginning a new financial calculation, it is good practice to clear all of the financial registers by pressing **f CLEAR FIN**. Frequently, however, you may want to repeat a calculation after changing a number in only one of the financial registers. To do so, do not press **f CLEAR FIN**; instead, simply store the new number in the register. The numbers in the other financial registers remain unchanged.

The financial registers are also cleared when you press **f CLEAR REG** and when Continuous Memory is reset. As described in Error Conditions.

The Cash Flow Sign Convention

When entering the **PV**, **PMT**, and **FV** cash flows, the quantities must be keyed into the calculator with proper sign, + (plus) or - (minus), in accordance with ...

The Cash Flow Sign Convention: Money received is entered or displayed as a positive value. Money paid out is entered or displayed as a negative value.

The Payment Mode

One more bit of information must be specified before you can solve a problem involving periodic payments. Such payments can be made either at the beginning of a compounding period (payments in advance, or annuities due) or at the end of the period (payments in arrears, or ordinary annuities). Calculations involving payments in advance yield different results than calculations involving payments in arrears. Regardless of whether payments are made in advance or in arrears, the number of payments must be the same as the number of compounding periods.

To specify the payment mode: press **g BEG** if payments are made at the beginning of the compounding period, press **g END** if payments are made at the end of the compounding period.

The **BEGIN** status indicator is lit when the payment mode is set to Begin. If **BEGIN** is not lit, the payment mode is set to End.

Compound Interest Calculations

Specifying the Number of Compounding Periods and the Periodic Interest Rate

Interest rates are usually quoted at the annual rate (also called the nominal rate): that is, the interest rate per year. However, in compound interest problems, the interest rate entered into **i** must always be expressed in terms of the basic compounding period, which may be years, months, days, or any other time unit. For example, if a problem involves 6% annual interest compounded quarterly for 5 years, **n**--the number of quarters--would be 5 x 4 = 20 and **i**--the interest rate per quarter--would be 6% / 4 = 1.5%. If the interest were instead compounded monthly, **n** would be 5 x 12 = 60 and **i** would be 6% / 12 = 0.5%.

Calculating the Number of Payments or Compounding Periods

Press **f CLEAR FIN** to clear the financial registers. Enter the periodic interest rate, using **i** or **12/**. Enter at least two of the following values: present value, using **PV**, payment amount, using **PMT**, future value, using **FV**. If a **PMT** was entered, press **g BEG** or **g END** to set the payment mode. Press **n** to calculate the number of payments or periods.

If the answer calculated is not an integer, the calculator rounds the answer up to the next higher integer before storing it in the **n** register and displaying it. For example, if **n** were calculated as 318.15, **319** would be the displayed answer. The value of **n** is rounded up by the calculator to show the total number of payments needed: **n**-1 equal, full payments, and one final, smaller payment.

Calculating the Periodic and Annual Interest Rates

Press **f CLEAR FIN** to clear the financial registers. Enter the number of payments or periods, using **n** or **12x**. Enter at least two of the following values: present value, using **PV**, payment amount, using **PMT**, future value, using **FV**. If a **PMT** was entered, press **g BEG** or **g END** to set the payment mode. Press **i** to calculate the periodic interest rate. To calculate the annual interest, key in the number of periods per year, then press **x**.

Calculating the Present Value

Press **f CLEAR FIN** to clear the financial registers. Enter the number of payments or periods, using **n** or **12x**. Enter the periodic interest rate, using **i** or **12/**. Enter either or both of the following: payment amount, using **PMT**, future value, using **FV**. If a **PMT** was entered, press **g BEG** or **g END** to set the payment mode. Press **PV** to calculate the present value.

Calculating the Payment Amount

Press **f CLEAR FIN** to clear the financial registers. Enter the number of payments or periods, using **n** or **12x**. Enter the periodic interest rate, using **i** or **12/**. Enter either or both of the following: present value, using **PV**, future value, using **FV**. Press **g BEG** or **g END** to set the payment mode. Press **PMT** to calculate the payment amount.

Calculating the Future Value

Press **f CLEAR FIN** to clear the financial registers. Enter the number of payments or periods, using **n** or **12x**. Enter the periodic interest rate, using **i** or **12/**. Enter either or both of the following: present value, using **PV**, payment amount, using **PMT**. If a **PMT** was entered, press **g BEG** or **g END** to set the payment mode. Press **FV** to calculate the future value.

Odd-Period Calculations

The examples presented so far have dealt with financial transactions in which interest begins to accrue at the beginning of the first regular payment period. However, interest often begins to accrue prior to the beginning of the first regular payment period. The period from the date interest begins accruing to the date of the first payment, begin not equal to the regular payment periods is sometimes referred to as an "odd first period." For simplicity, in using the HP-12C we will always regard the first period as equal to the remaining periods, and we will refer to the *period between the date interest begins accruing and the beginning of the first payment period* as simply the "odd period" or the "odd days". (Note that the odd period is always assumed by the calculator to occur before the first full payment period.) You can calculate **i**, **PV**, **PMT**, and **FV** for transactions involving an odd period simply by entering a *noninteger* **n**. This places the calculator in Odd-Period mode. The integer portion of **n** specifies the number of full payments periods, and the fractional part specifies the length of the odd period as a fraction of a full period.

At your option, the calculations of **i**, **PV**, **PMT**, and **FV** can be performed with either simple interest or compound interest accruing during the odd period. If the **C** status indicator in the display is not lit, simple interest is used. To toggle compound interest mode--turn the **C** indicator on or off--press **STO EEX**.

Programming Basics

Why Use Programs?

A program is simply a sequence of keystrokes that is stored in the calculator. Whenever you have to calculate with the same sequence of keystrokes several times, you can save a great deal of time by incorporating these keystrokes in a program. Instead of pressing all the keys each time, you press just one key to start the program: the calculator does the rest automatically!

Creating a Program

Creating a program consists simply of *writing* the program, then *storing* it: Write down the sequence of keystrokes that you would use to calculate the quantity or quantities desired. Press **f P/R** to set the calculator to Program mode. When the calculator is in Program mode, functions are not executed when they are keyed in, but instead are stored inside the calculator. The **PRGM** status indicator in the display is lit when the calculator is in Program mode.

Press **f CLEAR PRGM** to erase any previous programs that may be stored inside the calculator. If you want to create a new program without erasing a program already stored, then press **g GTO . nn** where **nn** is the program line of the last instruction of the previous program. (The last instruction of any program should be **g GTO 00**.) Begin keying in the new program. The new instructions are automatically stored starting after the last instruction of the previous program.

To run the program press **f P/R** to set the calculator back to Run mode. Key into the display or store into storage registers any data needed by the program. Press **R/S** to begin program execution.

Program Memory

Keystrokes entered into the calculator in Program mode are stored in *program memory*. Each digit, decimal point, or function key is called an *instruction* and is stored in one line of program memory--usually referred to simply as a *program line*. Keystroke sequences beginning with the **f**, **g**, **STO**, **RCL**, and **GTO** prefix keys are considered to comprise a *complete instruction* and are stored in only one program line.

When a program is run, each instruction in program memory is executed--that is, the keystroke in that program line is performed, just as if you were pressing the key manually--beginning with the current line in program memory and proceeding sequentially with the high-numbered program lines.

Whenever the calculator is in Program mode (**PRGM** status indicator is lit), the display shows information about the program line to which the calculator is currently set. At the left of the display is the number of the program line within program memory. The remaining digits in the display comprise a code that indicates what instruction has been stored in that program line. No code is shown for program line 00, since no regular instruction is stored there.

Identifying Instructions in Program Lines

Each key on the keyboard--except for the digit keys **0** through **9**--is identified by a two-digit "keycode" that corresponds to the key's position on the keyboard. The first digit in the keycode is the number of key row, counting from row 1 at the top; the second digit is the number of the key in that row, counting from 1 for the first key in the row through 9 for ninth key in the row and 0 for tenth key in the row. The keycode for each digit key is simply the digit on the key.

The GTO 00 Instruction and Program Line 00

Whenever you run a program, the last instruction should be **g GTO 00**. Thus, after each time a program is run, the calculator automatically goes to program line 00 and halts, ready for you to key in new data and run the program again. Whenever you press **f CLEAR PRGM** (in Program mode), the instruction **g GTO 00** is automatically stored in all program lines of program memory.

Setting the Calculator to a Particular Program Line

There will be occasions when you'll want to set the calculator directly to a particular program line--such as when you're storing a second program in program memory or when you're modifying an existing program. Although you can set the calculator to any line by using **SST**, you can do so more quickly as follows. In Program mode, pressing **g GTO .** followed by two digit keys sets the calculator to the program line specified. In Run mode, pressing **g GTO** followed by two digit keys sets the calculator to the program line specified by the digit keys. Since the calculator is not in Program mode, the line number and keycode are not displayed. The decimal point is not necessary if the calculator is in RUN mode, but it *is* necessary if the calculator is in Program mode.

Executing a Program One Line at a Time

Pressing **SST** while the calculator is in Run mode advances the calculator to the next line in program memory, then displays that line's number and the keycode of the instruction stored there, just as in Program mode. In Run mode, however, when the **SST** key is released the instruction in the program line just displayed is executed and the display then shows the result of executing that line.

Interrupting Program Execution

Occasionally you'll want a program to stop executing so that you can see an intermediate result or enter new data. The HP-12C provides two functions for doing so: **PSE** (*pause*) and **R/S** (*run/stop*). When a running program executes a **PSE** instruction, program execution pauses for about 1 second, then resumes. During the pause, the calculator displays the last result calculated before the **PSE** instruction was executed. When **R/S** is executed as a program instruction, the calculator halts program execution indefinitely, presumably to display some intermediate result. The user can then resume program execution at the next consecutive program line by pressing **R/S**.

If you press any key during a pause, program execution is halted indefinitely. To resume program execution at the program line following that containing the **PSE** instruction, press **R/S**.

Program execution is halted automatically when the calculator overflows or attempts an improper operation that results in an **Error** display. Either of these conditions signifies that the program itself probably contains an error.

To determine at which program line execution has halted (in order to locate the error), press any key to clear the **Error** display, then press **f P/R** to set the calculator to Program mode and display that program line.

You may also want to display the current program line (by pressing **f P/R**) if your program has halted at one of several **R/S** instructions in your program and you want to determine which one that is. To continue executing the program afterward: Press **f P/R** to set the calculator back to Run mode (remember that when switching from Program mode to Run mode the calculator is automatically set to program line 00). If you want to resume execution from the program line at which execution halted rather than from line 00, press **g GTO** followed by two digit keys that specify the program line desired (that program line should be the one after the **R/S** instruction). Press **R/S** to resume execution.

Branching and Looping

Although the instructions in a program normally are executed in order of their program line numbers, in some situations it is desirable to have program execution transfer or "branch" to a program line that is not the next line in program memory. Branching also makes it possible to automatically execute portions of a program more than once--a process called "looping".

There are two conditional test instructions that are used in programs for conditional branching: **x<=y** tests whether the number in the X-register is less than or equal to the number in the Y-register. **x=0** tests whether number in the X-register is equal to zero. If the conditional tests true, the calculator continues execution at the next program memory line. If false, the calculator skips the next line before resuming execution.

Bond Calculations

The HP-12C enables you to solve for bond price (and the interest accrued since the last interest date) and the yield to maturity. The **PRICE** and **YTM** calculations are done assuming a semiannual coupon payment and using an actual/actual basis (such as for U.S Treasury bonds and U.S. Treasury notes). In accordance with market convention, prices are based on a redemption (par) value of 100.

Calculating Bond Price

To calculate bond price: Enter the desired yield to maturity (as a percentage), using **i**. Enter the annual coupon rate (as a percentage), using **PMT**. Key in the settlement (purchase) date (as described in Keyboard Functions for **D.MY** or **M.DY**), then press **ENTER**. Key in the maturity (redemption) date. Press **f PRICE**.

The price is shown in the display and also is stored in the **PV** register. The interest accrued since the last interest date is held inside the calculator: to display the interest, press **x><y**; to add the interest to the price, press **+**.

Calculating Bond Yield

To calculate the yield to maturity: Enter the quoted price (as a percent of par), using **PV**. Enter the annual coupon rate (as a percentage), using **PMT**. Key in the settlement (purchase) date, then press **ENTER**. Key in the maturity (redemption) date. Press **f YTM**.

The yield to maturity is shown in the display and also stored in the **i** register.

Cash Flow Analysis

Discounted Cash Flow Analysis: NPV and IRR

The HP-12C provides functions for the two most widely-used methods of discounted cash flow analysis: **NPV** (net present value) and **IRR** (internal rate of return). These functions enable you to analyze financial problems involving cash flows (money paid out or recieved) occurring at regular intervals. As in compound interest calculations, the interval between cash flows can be any time period; however, the amounts of these cash flows need not be equal.

To understand how to use **NPV** and **IRR**, let's consider the cash flow for an investment that requires an initial cash outlay (**CF0**) and generates a cash flow (**CF1**) at the end of the first year, and so on up the final cash flow (**CFj**).

NPV is calculated by adding the initial investment (represented as a negative cash flow) to the present value of the anticipated future cash flows. The interest rate, **i**, will be referred to in this discussion of **NPV** and **IRR** as the *rate of return*. The value of **NPV** indicates the result of the investment: If **NPV** is positive, the financial value of the investor's assets would be increased--the investment is financially attractive. If **NPV** is zero, the financial value of the investor's assets would not change--the investor is indifferent toward the investment. If **NPV** is negative, the financial value of the investor's assets would be decreased--the investment is not financially attractive.

A comparison of the **NPV**'s of alternative investment possibilities indicates which of them is most desirable: the greater the **NPV**, the greater the increase in the financial value of the investor's assets.

IRR is the rate of return at which the discounted future cash flows equal the initial cash outlay: **IRR** is the discount rate at which **NPV** is zero. The value of **IRR** relative to the present value discount rate also indicates the result of the investment: If **IRR** is greater than the desired rate of return, the investment is financially attractive. If **IRR** is equal to the desired rate of return, the investor is indifferent toward the investment. If **IRR** is less than the desired rate of return, the investment is not financially attractive.

Calculating Net Present Value (NPV)

Calculating NPV for Ungrouped Cash Flows

Calculates the net present value of up to 20 uneven cash flows (in addition to the initial investment **CFo**). The amount of the the initial investment (**CFo**) is entered into the calculator using the **CFo** key. Pressing **g CFo** stores **CFo** in storage register R0 and also stores the number 0 in the **n** register.

The amounts of the subsequent cash flows are stored--in the order they occur--in the remaining storage registers: **CF1** thru **CF9** in R1 thru R9, and **CF10** thru **CF19** in R.0 thru R.9, respectively. If there is a **CF20**, that amount is stored in the **FV** register. Each of these cash flows (**CF1**, **CF2**, etc.) is designated **CFj**, where j takes on values from 1 up to the number of the final cash flow. The amounts of these cash flows are all entered using the **CFj** key. Each time **g CFj** is pressed, the amount in the display is stored in the next available storage register, and the number in the **n** register is increased by 1. This register therefore counts how many cash flow amounts (in addition to the initial investment **CFo**) have been entered.

When entering cash flow amounts--including the initial investment **CFo**--remember to observe the cash flow sign convention by pressing **CHS** after keying in a negative cash flow.

In summary, to enter the cash flow amounts: Press **f CLEAR REG** to clear the financial and

storage registers. Key in the amount of the initial investment, press **CHS** if that cash flow is negative, then press **g CFo**. If there is no initial investment, press **0 g CFo**. Key in the amount of the next cash flow, press **CHS** if the cash flow is negative, then press **g CFj**. If the cash flow amount is zero in the next period, press **0 g CFj**. Repeat last step for each cash flow until all have been entered. Enter the interest rate, using **i** or **12/**. Press **f NPV**.

The calculated value of **NPV** appears in the display and also is automatically stored in the **PV** register.

Calculating NPV for Grouped Cash Flows

A maximum of 20 cash flow *amounts* (in addition to initial investment **CFo**) can be stored in the HP-12C. However, problems involving more than 20 cash flows can be handled if among the cash flows there are *equal consecutive* cash flows. For such problems, you merely enter along with the amounts of the cash flows the number of times--up to 99--each amount occurrs consecutively. This number is designated **Nj**, corresponding to cash flow amount **CFj**, and is entered using the **Nj** key. Each **Nj** is stored in a special register inside the calculator.

This method can, of course, be used for problems involving fewer than 20 cash flows.

In summary, to enter the cash flow amounts: Press **f CLEAR REG** to clear the financial and storage registers. Key in the amount of the initial investment, press **CHS** if that cash flow is negative, then press **g CFo**. If there is no initial investment, press **0 g CFo**. If the initial investment consists of more than one cash flow of the amount entered, key in the number of those cash flows, then press **g Nj**. If **g Nj** is not pressed, the calculator assumes that **N0** is 1. Key in the amount of the next cash flow, press **CHS** if the cash flow is negative, then press **g CFj**. If the cash flow amount is zero in the next period, press **0 g CFj**. If the amount entered occurs more than once consecutively, key in the number of times that cash flow amount occurs consecutively, then press **g Nj**. If **g Nj** is not pressed, the calculator assumes that **Nj** is 1 for the **CFj** just entered. Repeat last step for each cash flow until all have been entered. Enter the interest rate, using **i** or **12/**. Press **f NPV**.

The calculated value of **NPV** appears in the display and also is automatically stored in the **PV** register.

Calculating Internal Rate of Return (IRR)

Enter the cash flows using the method described for **NPV**. Press **f IRR**. The calculated value of **IRR** appears in the display and also is automatically stored in the **i** register.

If you get an **Error 3** message in the display, you can try the procedure outlined in Error Conditions regarding an **Error 3** message to resolve this condition by entering a new guess for **IRR**.

Reviewing Cash Flow Entries

To display a single cash flow amount, press **RCL**, then key in the number of the register containing the cash flow amount to be displayed. Alternatively, store the number of that cash flow amount (that is, the value of j for the **CFj** desired) in the **n** register, then press **RCL g CFj**.

To review all the cash flow amounts, press **RCL g CFj** repeatedly. This displays the cash flow amounts in reverse order--that is, beginning with the final cash flow and proceeding to **CF0**.

To display the number of times a cash flow amount occurs consecutively--that is, to display the **Nj** for a **CFj**--store the number of that cash flow amount (that is, the value of j) in the **n** register, the press **RCL g Nj**.

To review all the cash flow amounts together with the number of times each cash flow amount

occurs consecutively (that is, to review each **CFj** and **Nj** pair), press **RCL g Nj RCL g CFj** repeatedly. This displays **Nj** followed by **CFj** beginning with the final cash flow amount and proceeding to **N0** and **CF0**.