

robodoc

J.v.Weert and F.N.C.Slothouber

| |
|----------------------|
| COLLABORATORS |
|----------------------|

| | | | |
|---------------|-----------------------------------|------------------|------------------|
| | <i>TITLE :</i> robodoc | | |
| <i>ACTION</i> | <i>NAME</i> | <i>DATE</i> | <i>SIGNATURE</i> |
| WRITTEN BY | J.v.Weert and F.N.C.Slothouber | December 8, 2024 | |

| |
|-------------------------|
| REVISION HISTORY |
|-------------------------|

| | | | |
|--------|------|-------------|------|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

Contents

| | | |
|----------|---|----------|
| 1 | robodoc | 1 |
| 1.1 | DateClass.guide | 1 |
| 1.2 | DateClass/--background-- | 3 |
| 1.3 | DateClass/--compiling-- | 6 |
| 1.4 | DateClass/--history-- | 6 |
| 1.5 | DateClass/--todo-- | 8 |
| 1.6 | DateClass/Easter() | 8 |
| 1.7 | DateClass/date | 9 |
| 1.8 | DateClass/date/date() | 10 |
| 1.9 | DateClass/date/date() | 10 |
| 1.10 | DateClass/date/getday() | 11 |
| 1.11 | DateClass/date/getjd() | 12 |
| 1.12 | DateClass/date/getmjd() | 13 |
| 1.13 | DateClass/date/getmonth() | 14 |
| 1.14 | DateClass/date/getmoonage() | 15 |
| 1.15 | DateClass/date/getmoonphase() | 15 |
| 1.16 | DateClass/date/getscaliger() | 16 |
| 1.17 | DateClass/date/getweek() | 17 |
| 1.18 | DateClass/date/getweekday() | 18 |
| 1.19 | DateClass/date/getyear() | 19 |
| 1.20 | DateClass/date/operator!() | 20 |
| 1.21 | DateClass/date/operator!==(int) | 20 |
| 1.22 | DateClass/date/operator!==(int) | 21 |
| 1.23 | DateClass/date/operator+() | 22 |
| 1.24 | DateClass/date/operator+() | 23 |
| 1.25 | DateClass/date/operator++() | 24 |
| 1.26 | DateClass/date/operator++(int) | 25 |
| 1.27 | DateClass/date/operator+==(int) | 26 |
| 1.28 | DateClass/date/operator+==(int) | 26 |
| 1.29 | DateClass/date/operator-() | 27 |

| | | |
|------|--|----|
| 1.30 | DateClass/date/operator-() | 28 |
| 1.31 | DateClass/date/operator-() | 29 |
| 1.32 | DateClass/date/operator--() | 30 |
| 1.33 | DateClass/date/operator--(int) | 31 |
| 1.34 | DateClass/date/operator-=() | 31 |
| 1.35 | DateClass/date/operator-=() | 32 |
| 1.36 | DateClass/date/operator<() | 33 |
| 1.37 | DateClass/date/operator<() | 34 |
| 1.38 | DateClass/date/operator<=() | 35 |
| 1.39 | DateClass/date/operator<=() | 36 |
| 1.40 | DateClass/date/operator==() | 37 |
| 1.41 | DateClass/date/operator==() | 38 |
| 1.42 | DateClass/date/operator>() | 39 |
| 1.43 | DateClass/date/operator>() | 39 |
| 1.44 | DateClass/date/operator>=() | 40 |
| 1.45 | DateClass/date/operator>=() | 41 |
| 1.46 | DateClass/dateduration | 42 |
| 1.47 | DateClass/dateduration/dateduration() | 43 |
| 1.48 | DateClass/dateduration/getdays() | 43 |
| 1.49 | DateClass/dateduration/getindays() | 44 |
| 1.50 | DateClass/dateduration/getmonths() | 45 |
| 1.51 | DateClass/dateduration/getyears() | 46 |
| 1.52 | DateClass/dateduration/operator!=() | 47 |
| 1.53 | DateClass/dateduration/operator%() | 48 |
| 1.54 | DateClass/dateduration/operator*() | 48 |
| 1.55 | DateClass/dateduration/operator*() | 49 |
| 1.56 | DateClass/dateduration/operator*() | 50 |
| 1.57 | DateClass/dateduration/operator*=() | 51 |
| 1.58 | DateClass/dateduration/operator*=() | 52 |
| 1.59 | DateClass/dateduration/operator*=() | 53 |
| 1.60 | DateClass/dateduration/operator+() | 54 |
| 1.61 | DateClass/dateduration/operator++() | 54 |
| 1.62 | DateClass/dateduration/operator++(int) | 55 |
| 1.63 | DateClass/dateduration/operator+=() | 56 |
| 1.64 | DateClass/dateduration/operator-() | 57 |
| 1.65 | DateClass/dateduration/operator--() | 58 |
| 1.66 | DateClass/dateduration/operator--(int) | 59 |
| 1.67 | DateClass/dateduration/operator-=() | 59 |
| 1.68 | DateClass/dateduration/operator/() | 60 |

| | | |
|-------|--------------------------------------|----|
| 1.69 | DateClass/dateduration/operator/() | 61 |
| 1.70 | DateClass/dateduration/operator/() | 62 |
| 1.71 | DateClass/dateduration/operator/() | 63 |
| 1.72 | DateClass/dateduration/operator/=() | 64 |
| 1.73 | DateClass/dateduration/operator/=() | 65 |
| 1.74 | DateClass/dateduration/operator/=() | 66 |
| 1.75 | DateClass/dateduration/operator<=() | 66 |
| 1.76 | DateClass/dateduration/operator==() | 67 |
| 1.77 | DateClass/dateduration/operator>() | 68 |
| 1.78 | DateClass/dateduration/operator>=() | 69 |
| 1.79 | DateClass/dateeduration/operator<() | 70 |
| 1.80 | DateClass/time | 71 |
| 1.81 | DateClass/time/gethour() | 71 |
| 1.82 | DateClass/time/getjd() | 72 |
| 1.83 | DateClass/time/getmin() | 73 |
| 1.84 | DateClass/time/getsec() | 74 |
| 1.85 | DateClass/time/getzone() | 75 |
| 1.86 | DateClass/time/operator!=() | 76 |
| 1.87 | DateClass/time/operator()() | 76 |
| 1.88 | DateClass/time/operator+() | 77 |
| 1.89 | DateClass/time/operator+=() | 78 |
| 1.90 | DateClass/time/operator-() | 79 |
| 1.91 | DateClass/time/operator-() | 80 |
| 1.92 | DateClass/time/operator=() | 81 |
| 1.93 | DateClass/time/operator<() | 82 |
| 1.94 | DateClass/time/operator<<() | 82 |
| 1.95 | DateClass/time/operator<<=() | 83 |
| 1.96 | DateClass/time/operator<=() | 84 |
| 1.97 | DateClass/time/operator==() | 85 |
| 1.98 | DateClass/time/operator>() | 86 |
| 1.99 | DateClass/time/operator>=() | 87 |
| 1.100 | DateClass/time/operator>>() | 88 |
| 1.101 | DateClass/time/operator>>=() | 89 |
| 1.102 | DateClass/time/operator[]() | 89 |
| 1.103 | DateClass/time/time() | 90 |
| 1.104 | DateClass/time/time() | 91 |
| 1.105 | DateClass/time/time() | 92 |
| 1.106 | DateClass/timeduration | 93 |
| 1.107 | DateClass/timeduration/gethours() | 93 |

| | | |
|-------|---------------------------------------|-----|
| 1.108 | DateClass/timeduration/getinsecs() | 94 |
| 1.109 | DateClass/timeduration/getmins() | 95 |
| 1.110 | DateClass/timeduration/getsecs() | 96 |
| 1.111 | DateClass/timeduration/operator!=() | 97 |
| 1.112 | DateClass/timeduration/operator%() | 97 |
| 1.113 | DateClass/timeduration/operator*() | 98 |
| 1.114 | DateClass/timeduration/operator*() | 99 |
| 1.115 | DateClass/timeduration/operator*() | 100 |
| 1.116 | DateClass/timeduration/operator*=() | 101 |
| 1.117 | DateClass/timeduration/operator*=() | 102 |
| 1.118 | DateClass/timeduration/operator*=() | 103 |
| 1.119 | DateClass/timeduration/operator+() | 104 |
| 1.120 | DateClass/timeduration/operator+=() | 104 |
| 1.121 | DateClass/timeduration/operator-() | 105 |
| 1.122 | DateClass/timeduration/operator-=() | 106 |
| 1.123 | DateClass/timeduration/operator/() | 107 |
| 1.124 | DateClass/timeduration/operator/() | 108 |
| 1.125 | DateClass/timeduration/operator/() | 109 |
| 1.126 | DateClass/timeduration/operator/() | 110 |
| 1.127 | DateClass/timeduration/operator/=(()) | 111 |
| 1.128 | DateClass/timeduration/operator/=(()) | 111 |
| 1.129 | DateClass/timeduration/operator/=(()) | 112 |
| 1.130 | DateClass/timeduration/operator<() | 113 |
| 1.131 | DateClass/timeduration/operator<=() | 114 |
| 1.132 | DateClass/timeduration/operator==(()) | 115 |
| 1.133 | DateClass/timeduration/operator>() | 116 |
| 1.134 | DateClass/timeduration/operator>=() | 117 |
| 1.135 | DateClass/timeduration/timeduration() | 118 |
| 1.136 | DateClass/timeduration/timeduration() | 118 |
| 1.137 | DateClass/timeduration/timeduration() | 119 |

Chapter 1

robodoc

1.1 DateClass.guide

TABLE OF CONTENTS

DateClass/--background--
DateClass/--compiling--
DateClass/--history--
DateClass/--todo--
DateClass/Easter()
DateClass/date
DateClass/date/date()
DateClass/date/date()
DateClass/date/getday()
DateClass/date/getjd()
DateClass/date/getmjd()
DateClass/date/getmonth()
DateClass/date/getmoonage()
DateClass/date/getmoonphase()
DateClass/date/getscaliger()
DateClass/date/getweek()
DateClass/date/getweekday()
DateClass/date/getyear()
DateClass/date/operator!()
DateClass/date/operator!=()
DateClass/date/operator!=()
DateClass/date/operator+()
DateClass/date/operator+()
DateClass/date/operator++()
DateClass/date/operator++(int)
DateClass/date/operator+=()
DateClass/date/operator+=()
DateClass/date/operator-()
DateClass/date/operator-()
DateClass/date/operator-()
DateClass/date/operator--()
DateClass/date/operator--(int)
DateClass/date/operator-=()
DateClass/date/operator-=()
DateClass/date/operator<()
DateClass/date/operator<()
DateClass/date/operator<=()

```
DateClass/date/operator<= ()
DateClass/date/operator== ()
DateClass/date/operator== ()
DateClass/date/operator> ()
DateClass/date/operator> ()
DateClass/date/operator>= ()
DateClass/date/operator>= ()
DateClass/date/struct
DateClass/dateduration
DateClass/dateduration/dateduration ()
DateClass/dateduration/getdays ()
DateClass/dateduration/getindays ()
DateClass/dateduration/getmonths ()
DateClass/dateduration/getyears ()
DateClass/dateduration/operator!= ()
DateClass/dateduration/operator% ()
DateClass/dateduration/operator* ()
DateClass/dateduration/operator* ()
DateClass/dateduration/operator* ()
DateClass/dateduration/operator* ()
DateClass/dateduration/operator*= ()
DateClass/dateduration/operator*= ()
DateClass/dateduration/operator*= ()
DateClass/dateduration/operator+ ()
DateClass/dateduration/operator++ ()
DateClass/dateduration/operator++ (int)
DateClass/dateduration/operator+= ()
DateClass/dateduration/operator- ()
DateClass/dateduration/operator-- ()
DateClass/dateduration/operator-- (int)
DateClass/dateduration/operator-= ()
DateClass/dateduration/operator/ ()
DateClass/dateduration/operator/ ()
DateClass/dateduration/operator/ ()
DateClass/dateduration/operator/ ()
DateClass/dateduration/operator/= ()
DateClass/dateduration/operator/= ()
DateClass/dateduration/operator/= ()
DateClass/dateduration/operator<= ()
DateClass/dateduration/operator== ()
DateClass/dateduration/operator> ()
DateClass/dateduration/operator>= ()
DateClass/dateduration/struct
DateClass/dateeduration/operator< ()
DateClass/time
DateClass/time/gethour ()
DateClass/time/getjd ()
DateClass/time/getmin ()
DateClass/time/getsec ()
DateClass/time/getzone ()
DateClass/time/operator!= ()
DateClass/time/operator () ()
DateClass/time/operator+ ()
DateClass/time/operator+= ()
DateClass/time/operator- ()
DateClass/time/operator- ()
DateClass/time/operator-= ()
DateClass/time/operator< ()
```

```

DateClass/time/operator<<()
DateClass/time/operator<<=()
DateClass/time/operator<=()
DateClass/time/operator==()
DateClass/time/operator>()
DateClass/time/operator>=()
DateClass/time/operator>>()
DateClass/time/operator>>=()
DateClass/time/operator[]()
DateClass/time/struct
DateClass/time/time()
DateClass/time/time()
DateClass/time/time()
DateClass/timeduration
DateClass/timeduration/gethours()
DateClass/timeduration/getinsecs()
DateClass/timeduration/getmins()
DateClass/timeduration/getsecs()
DateClass/timeduration/operator!=()
DateClass/timeduration/operator%()
DateClass/timeduration/operator*()
DateClass/timeduration/operator*()
DateClass/timeduration/operator*()
DateClass/timeduration/operator*()
DateClass/timeduration/operator*()
DateClass/timeduration/operator*()
DateClass/timeduration/operator*()
DateClass/timeduration/operator+()
DateClass/timeduration/operator+=()
DateClass/timeduration/operator-()
DateClass/timeduration/operator-=()
DateClass/timeduration/operator/()
DateClass/timeduration/operator/()
DateClass/timeduration/operator/()
DateClass/timeduration/operator/()
DateClass/timeduration/operator/=()
DateClass/timeduration/operator/=()
DateClass/timeduration/operator/=()
DateClass/timeduration/operator<()
DateClass/timeduration/operator<=()
DateClass/timeduration/operator==()
DateClass/timeduration/operator>()
DateClass/timeduration/operator>=()
DateClass/timeduration/struct
DateClass/timeduration/timeduration()
DateClass/timeduration/timeduration()
DateClass/timeduration/timeduration()
DateClass/timeduration/timeduration()

```

1.2 DateClass/--background--

DateClass/--background--

NAME

DateClass -- This class was designed to help calc. cal. dates (V33)

FUNCTION

This class has been designed to become a useful and portable class and to help developers handle calendar dates!

NOTES

This class is based on the date library copyrighted 1994-95 by Kai Hofmann.

COPYRIGHT

This software is copyrighted 1995 by Kai Hofmann.
All rights reserved!

- Permission for COMMERCIAL USE is only given by an extra available commercial license that must be validated!
Contact me directly for this license, because it will be individually handed out per your needs!
- Permission is hereby granted, without written agreement and without license, to USE this software and its documentation for any NON-COMMERCIAL purpose, provided that the above copyright notice and the following paragraph appear in all copies of this software (Non-commercial includes Giftware and Shareware!).

You **must** include the following notice in your product and in your documentation:

"This software uses the date class that is copyrighted 1994-95 by Kai Hofmann"

You "must" send me a full version of your product at no cost including free updates!
Extra money is welcome (For Bank Account see below - but **ONLY** send in DM to this Bank Account!).

- THERE IS **NO PERMISSION** GIVEN TO MODIFY THIS SOFTWARE!

If you need only parts of this software, you should not worry, because it's the job of the optimizer from your C compiler to include only the needed parts in your executable!

DISCLAIMER

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDER AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,

INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE AUTHOR HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

DISTRIBUTION

Permission is hereby granted, without written agreement and without license or royalty fees, to copy and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following paragraphs appear in all copies of this software, to:

- All who will distribute this software for free!
- All free accessible INTERNET servers and PHONE boxes!
- All Aminet sites
- All SimTel sites
- Fred Fish for his great Amiga-Software-Library
- The German SAAR AG PD-Library
- All others who do NOT take more than \$5.- for one disk that includes this software!
- ALL others who do NOT take more than \$40.- for one CD that includes this software!

ADDITIONAL INFORMATION

I have tried to make portable/useful and I hope bugfree software for eternity - but this seems to be impossible (sorry!) :)
So I hope you will pay a fee for this.

Kindly send US - dollars to a friend of mine in the USA who will forward it to me in a timely manner. Please send checks or money orders only.
Contact me via email for more!

AUTHOR

Kai Hofmann
Arberger Heerstraße 92
28307 Bremen
Germany

Phone: (+49)-(0)421/480780
(Remember that my parents don't speak english!)

E-Mail: i07m@zfn.uni-bremen.de
i07m@informatik.uni-bremen.de

IRC : PowerStat@#AmigaGer

WWW : <http://www.informatik.uni-bremen.de/~i07m>

Bank account : 1203 7503
Account owner: Kai Hofmann
Bank code : 290 501 01
Bank name : Sparkasse in Bremen/Germany

THANKS

Many thanks to the following people:

- | | |
|---|---|
| Rita Reichl | - For correcting my rusty english (very often) |
| Christian Schaefer | - For spending time on this class with his Borland C++ 4.5 compiler |
| Jacco van Weert & Frans Slothouber | - For the 'Robodoc' utility |
| James Cooper and the other Amiga people at SAS Intitute | - For spending their unpaid free time with continuation of the Amiga SAS C/C++ support :) |

1.3 DateClass/--compiling--

DateClass/--compiling--

NAME

compiling -- Specials for compiling the date class.

COMPILING

- Compile the date library as C++ !!!
- You could compile this code as normal C++
- `_ISO8859_Latin1` should be defined if your system supports this font encoding technology!
- `Date.h` will be included automatically!

1.4 DateClass/--history--

DateClass/--history--

NAME

history -- This is the development history of the Date class

VERSION

\$VER: DateClass 33.050 (30.07.95)

HISTORY

- | | |
|--------------|---|
| 26.05.1995 - | Initialization of the following class: timeduration Implementation of timeduration constructors, operator==,operator!=,operator==,operator!=, operator<,operator>,operator<=,operator>=,operator+, operator-,operator+=,operator-= |
| 27.05.1995 - | Initialization of the following class: |

- dateduration
- Implementation of dateduration constructors
- 30.05.1995 - Implementation of operator==, operator!=, operator<, operator>, operator<=, operator>=, operator++, operator--, operator+, operator-, operator+=, operator-=, operator*, operator/, operator*=, operator/= for dateduration class
- Implementation of gethours(), getmins(), getsecs() for timeduration class
- 02.06.1995 - Fixing small bugs in timeduration constructors, operator-=
- 03.06.1995 - Implementing getdays(), getmonths(), getyears() for dateduration
- Removing one of the dateduration constructors
- Fixing small bugs in operator<, operator>, operator<=, operator>=
- 03.07.1995 - Initialization of the following class:
time
- 07.07.1995 - Initialization of the following class:
date
- Implementation of time constructors, gethour(), getmin(), getsec(), getzone(), getjd(), operator==, operator!=, operator<, operator>, operator<=, operator>=, operator-, operator+, operator-, operator+=, operator-=
- Implementation of date constructors, getday(), getmonth(), getyear(), getweek(), getweekday(), getjd(), getmjd(), getscliger(), operator==, operator!=, operator<, operator>, operator<=, operator>=, operator==, operator!=, operator<, operator>, operator<=, operator>=, operator!, operator-, operator+, operator-, operator+=, operator-=, operator+, operator-, operator+=, operator-=, changetozone(), operator<<, operator>>, operator<<=, operator>>=, operator[], operator()
- Implementation of timeduration getinsecs()
- Implementation of dateduration getindays()
- Optimizations
- 08.07.1995 - Fixing bugs in time
operator<<, operator>>, operator<<=, operator>>=, operator[], opertaor(), changetozone(), operator-, operator+, operator-
- 09.07.1995 - I found a BUG in the SAS C++ translator, and so I lost all the source from the DateClass.cpp :(Completely reconstructed in 8 hours!
- 10.07.1995 - Start to writing autodocs
- 14.07.1995 - Workaround for the SAS C++ enum problem in Date.c
- 15.07.1995 - Continuing with writing autodocs
- 16.07.1995 - Continuing with writing autodocs
- 17.07.1995 - Finishing with writing autodocs
- 18.07.1995 - Fixing conversion warnings.
- 27.07.1995 - Correcting my rusty English in the Autodocs.
- Implementing of timeduration operators:
operator*, operator/, operator*=, operator/=
- 28.07.1995 - Shortening the year of the version-string to 95, because the Amiga 'version' command is buggy!

```

                Removing the changetozone() function from the time
                class, because it's better to use the operator[].
                Starting to write a test program.
29.07.1995 -    Implementing Easter()
                Implementing of date functions:
                getmoonage(),getmoonphase()
                Implementing of timeduration operator:
                operator/()
                Implementing of dateduration operator:
                operator/()
30.07.1995 -    Implementing of timeduration operator:
                operator%()
                Implementing of dateduration operator:
                operator%()
                Fixing bugs (Thanks to Christian Schaefer for spending
                time on this class with Borland C++ 4.5)

```

1.5 DateClass/--todo--

DateClass/--todo--

NAME

```
todo -- This is the 'To-Do' list of the Date class
```

TODO

```

Writing a test programm
Writing code for string operations/TextEngine
Derive a datetime class
Derive a datetimeduration class
Variable durations (2-3h)
Variable time points (18. or 19.3.)
Alternatives
etc. (that's top secret ;-)
```

1.6 DateClass/Easter()

DateClass/Easter()

NAME

```
Easter() -- Returns the date of Easter in a year (V33)
```

SYNOPSIS

```

date = Easter(year);

date Easter(const int year);
```

FUNCTION

Returns the date of Easter for a specified year.

INPUTS

year - Easter is calculated for this year

RESULT

date - Complete date of Easter Sunday

EXAMPLE

```
...  
date = Easter(1995);  
...
```

NOTES

Use this only for 31 to 2099!

BUGS

Unknown.

SEE ALSO

date/HeisEaster(), date

1.7 DateClass/date

DateClass/date

NAME

date -- Class for dates. (V33)

FUNCTION

This class maps dates.

NOTES

None.

BUGS

No known bugs.

SEE ALSO

timeduration, dateduration, time

1.8 DateClass/date/date()

DateClass/date/date()

NAME

date() -- Constructor of date class (V33)

SYNOPSIS

```
d = date(jd);  
  
date(unsigned long jd);
```

FUNCTION

Constructor of the date class for JD format.

INPUTS

jd - Date in JD format

RESULT

d - Date object

EXAMPLE

```
...  
date d(2449354);  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

date(), Date/JDToHeis()

1.9 DateClass/date/date()

DateClass/date/date()

NAME

date() -- Constructor of date class (V33)

SYNOPSIS

```
d = date(day,month,year);  
  
date(unsigned short day, unsigned short month, int year);
```

FUNCTION

Constructor of the date class.

INPUTS

day - Day of date
month - Month of date
year - Year of date

RESULT

d - Date object

EXAMPLE

```
...  
date d(17,7,1995);  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

date()

1.10 DateClass/date/getday()

DateClass/date/getday()

NAME

getday() -- Get day of date object (V33)

SYNOPSIS

```
day = (*this).getday();  
  
unsigned short getday(void) const;
```

FUNCTION

Get the day from the date object.

INPUTS

*this - date object

RESULT

day - Day of the date object

EXAMPLE

```
...
day = d.getday();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

getmonth(), getyear(), getweek(), getweekday(), getjd(), getmjd(),
getscaliger()

1.11 DateClass/date/getjd()

DateClass/date/getjd()

NAME

getjd() -- Get date in JD format from date object (V33)

SYNOPSIS

```
jd = (*this).getjd();

unsigned long getjd(void) const;
```

FUNCTION

Get the date from the date object in JD format.

INPUTS

*this - date object

RESULT

jd - Date of the date object in JD format

EXAMPLE

```
...
```

```
jd = d.getjd();  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
getday(), getmonth(), getyear(), getweek(), getweekday(), getmjd(),  
getscaliger(), Date/HeisToJD()
```

1.12 DateClass/date/getmjd()

DateClass/date/getmjd()

NAME

getmjd() -- Get date in MJD format from date object (V33)

SYNOPSIS

```
mjd = (*this).getmjd();  
  
unsigned long getmjd(void) const;
```

FUNCTION

Get the date from the date object in MJD format.

INPUTS

*this - date object

RESULT

mjd - Date of the date object in MJD format

EXAMPLE

```
...  
mjd = d.getmjd();  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`getday()`, `getmonth()`, `getyear()`, `getweek()`, `getweekday()`, `getjd()`,
`getscaliger()`, `Date/JDtoMJD()`, `Date/HeisToJD()`

1.13 DateClass/date/getmonth()

DateClass/date/getmonth()

NAME

`getmonth()` -- Get month of date object (V33)

SYNOPSIS

```
month = (*this).getmonth();  
  
unsigned short getmonth(void) const;
```

FUNCTION

Get the month from the date object.

INPUTS

`*this` - date object

RESULT

month - Month of the date object

EXAMPLE

```
...  
month = d.getmonth();  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`getday()`, `getyear()`, `getweek()`, `getweekday()`, `getjd()`, `getmjd()`,
`getscaliger()`

1.14 DateClass/date/getmoonage()

DateClass/date/getmoonage()

NAME

getmoonage() -- Returns the age of the moon (V33)

SYNOPSIS

```
ep = (*this).getmoonage();  
  
unsigned short getmoonage(void) const;
```

FUNCTION

Returns the age of the moon on a specified date.

INPUTS

*this - date object

RESULT

ep - The age of the moon on the specified date.

EXAMPLE

```
...  
ep = d.getmoonage();  
...
```

NOTES

Use this only for 1582 to 4100!
This is only a experimental version!

BUGS

Unknown

SEE ALSO

date/GregorianMoonAge()

1.15 DateClass/date/getmoonphase()

DateClass/date/getmoonphase()

NAME

getmoonphase() -- Searches for the next moon phase (V33)

SYNOPSIS

```
date = (*this).getmoonphase(phase);  
  
date getmoonphase(const MoonPhases phase) const;
```

FUNCTION

Returns the next moon phase you are searching for after a specified date.

INPUTS

*this - date object

RESULT

date - The date on which the moon phase was found.

EXAMPLE

```
...  
date = d.getmoonphase(FullMoon);  
...
```

NOTES

The range of this function is unknown to me!
So use it only from 1583 to 2500.
This is only an experimental version!

BUGS

Unknown

SEE ALSO

date/GregorianMoonPhase(), date/JDToHeis(), date()

1.16 DateClass/date/getscaliger()

DateClass/date/getscaliger()

NAME

getscaliger() -- Get year in Scaliger format from date object (V33)

SYNOPSIS

```
scaliger = (*this).getscaliger();  
  
unsigned int getscaliger(void) const;
```

FUNCTION

Get the year of the date object in Scaliger format.

INPUTS

*this - date object

RESULT

scaliger - Year of the date object in Scaliger format

EXAMPLE

```
...
syear = d.getscaliger();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

getday(), getmonth(), getyear(), getweek(), getweekday(), getjd(), getmjd(),
Date/HYearToScaliger()

1.17 DateClass/date/getweek()

DateClass/date/getweek()

NAME

getweek() -- Get week of date object (V33)

SYNOPSIS

```
week = (*this).getweek();

unsigned short getweek(void) const;
```

FUNCTION

Get the week from the date object.

INPUTS

*this - date object

RESULT

week - Week of the date object

EXAMPLE

```
...
week = d.getweek();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`getday()`, `getmonth()`, `getyear()`, `getweekday()`, `getjd()`, `getmjd()`,
`getscaliger()`, `Date/HeisWeek()`

1.18 DateClass/date/getweekday()

DateClass/date/getweekday()

NAME

`getweekday()` -- Get weekday of date object (V33)

SYNOPSIS

```
weekday = (*this).getweekday();

Weekdays getweekday(void) const;
```

FUNCTION

Get the weekday from the date object.

INPUTS

*this - date object

RESULT

weekday - Weekday of the date object

EXAMPLE

```
...
wd = d.getweekday();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`getday()`, `getmonth()`, `getyear()`, `getweek()`, `getjd()`, `getmjd()`,
`getscaliger()`, `Date/HeisWeekday()`

1.19 DateClass/date/getyear()

`DateClass/date/getyear()`

NAME

`getyear()` -- Get year of date object (V33)

SYNOPSIS

```
year = (*this).getyear();  
  
int getyear(void) const;
```

FUNCTION

Get the year from the date object.

INPUTS

`*this` - date object

RESULT

year - Year of the date object

EXAMPLE

```
...  
year = d.getyear();  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`getday()`, `getmonth()`, `getweek()`, `getweekday()`, `getjd()`, `getmjd()`,
`getscaliger()`

1.20 DateClass/date/operator!()

DateClass/date/operator!()

NAME

operator!() -- Leapyear operator for date object (V33)

SYNOPSIS

```
leapyear = !(*this);  
  
bool operator!() const;
```

FUNCTION

Checks if the year of the date object is a leap year.

INPUTS

*this - date object

RESULT

leapyear - Boolean, TRUE if the year of the date object is a leapyear,
FALSE if it is a normal year.

EXAMPLE

```
...  
if (!d)  
    printf("leap year\n");  
else  
    printf("normal year\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

Date/HeisLeapYear()

1.21 DateClass/date/operator!==()

DateClass/date/operator!==()

NAME

operator!=() -- Unequal operator for date objects (V33)

SYNOPSIS

```
compare = ((*this) != d);

bool operator!=(const date &d) const;
```

FUNCTION

Compares two date objects for unequality.

INPUTS

```
*this - date object
d      - date object
```

RESULT

compare - Boolean result of the unequal compare operation

EXAMPLE

```
...
if (d1 != d2)
    printf("!=\n");
else
    printf("==\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator==(), operator<(), operator>(), operator<=(), operator>=(),
Date/Compare2Dates()

1.22 DateClass/date/operator!=()

DateClass/date/operator!=()

NAME

operator!=() -- Unequal operator for date object and weekday (V33)

SYNOPSIS

```
compare = ((*this) != wd);
```

```
bool operator!=(const Weekdays wd) const;
```

FUNCTION

Compares if the date object is unequal to a weekday.

INPUTS

```
*this - date object  
wd    - Weekday
```

RESULT

compare - Boolean result of the unequal compare operation

EXAMPLE

```
...  
if (d1 != wd)  
    printf("!=\n");  
else  
    printf("==\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator==( ), operator<( ), operator>( ), operator<=( ), operator>=( ),  
Date/HeisWeekday( )
```

1.23 DateClass/date/operator+()

DateClass/date/operator+()

NAME

operator+() -- Add operator for date and dateduration objects (V33)

SYNOPSIS

```
d = (*this) + dd;
```

```
date operator+(const dateduration &dd) const;
```

FUNCTION

Add a date object and a dateduration object to a new date object.

INPUTS

```
*this - date object
dd    - dateduration object
```

RESULT

```
d - New date object
```

EXAMPLE

```
...
d1 = d2 + dd;
...
```

NOTES

```
None.
```

BUGS

```
The dateduration may not be longer (in days) than an integer value!
```

SEE ALSO

```
operator-(), operator+(), operator-(), date(), dateduration/getindays(),
Date/HeisDiffDate()
```

1.24 DateClass/date/operator+()

```
DateClass/date/operator+()
```

NAME

```
operator+() -- Add operator for date object and weekday (V33)
```

SYNOPSIS

```
d = (*this) + wd;

date operator+(const Weekdays wd) const;
```

FUNCTION

```
Gives you the next date, that is the specified weekday (could be the
same date!).
```

INPUTS

```
*this - date object
wd    - Weekday
```

RESULT

d - New date object

EXAMPLE

```
...  
d1 = d2 + wd;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator-()`, `operator+=()`, `operator-=()`, `date()`,
`Date/HeisDaysAfterWeekday()`, `Date/HeisDiffDate()`

1.25 DateClass/date/operator++()

DateClass/date/operator++()

NAME

`operator++()` -- Increment operator for date object (V33)

SYNOPSIS

```
d++;  
  
date operator++();
```

FUNCTION

Increments the date object by one day.

INPUTS

*this - date object

RESULT

*this - Modified date object

EXAMPLE

```
...  
d++;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator++(int)`, `operator--()`, `operator--(int)`, `Date/HeisDiffDate()`

1.26 DateClass/date/operator++(int)

`DateClass/date/operator++(int)`

NAME

`operator++(int)` -- Increment operator for date object (V33)

SYNOPSIS

`++d;`

`date operator++(int);`

FUNCTION

Increments the date object by one day.

INPUTS

`*this` - date object

RESULT

`*this` - Modified date object

EXAMPLE

```
...
++d;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator++()`, `operator--()`, `operator--(int)`, `Date/HeisDiffDate()`

1.27 DateClass/date/operator+=()

DateClass/date/operator+=()

NAME

operator+=() -- Add to operator for date and dateduration objects (V33)

SYNOPSIS

```
(*this) += dd;  
  
const date &operator+=(const dateduration &dd);
```

FUNCTION

Add a dateduration object to a date object.

INPUTS

*this - date object
dd - dateduration object

RESULT

*this - Modified date object

EXAMPLE

```
...  
d += dd;  
...
```

NOTES

None.

BUGS

The dateduration may not be longer (in days) than an integer value!

SEE ALSO

operator+(), operator-(), operator-=(), dateduration/getindays(),
Date/HeisDiffDate()

1.28 DateClass/date/operator+=()

DateClass/date/operator+=()

NAME

operator+=() -- Add to operator for date object and weekday (V33)

SYNOPSIS

```
(*this) += wd;

const date &operator+=(const Weekdays wd);
```

FUNCTION

Corrects the date object to the next date that is the specified weekday (could be the same date!).

INPUTS

```
*this - date object
wd    - Weekday
```

RESULT

```
*this - Modified date object
```

EXAMPLE

```
...
d += wd;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator+(), operator-(), operator-=(), Date/HeisDaysAfterWeekday(),
Date/HeisDiffDate()
```

1.29 DateClass/date/operator-()

DateClass/date/operator-()

NAME

operator-() -- Sub operator for date objects (V33)

SYNOPSIS

```
dd = (*this) - d;

dateduration operator-(const date &d) const;
```

FUNCTION

Subtract date object 2 from date object 1 to a new dateduration object.

INPUTS

```
*this - date object
d      - date object
```

RESULT

```
dd - New dateduration object
```

EXAMPLE

```
...
dd = d1 - d2;
...
```

NOTES

```
None.
```

BUGS

```
No known bugs.
```

SEE ALSO

```
operator+(), operator-(), dateduration(), Date/HeisDayDiff()
```

1.30 DateClass/date/operator-()

```
DateClass/date/operator-()
```

NAME

```
operator-() -- Sub operator for date and dateduration objects (V33)
```

SYNOPSIS

```
d = (*this) - dd;

date operator-(const dateduration &dd) const;
```

FUNCTION

```
Subtract a dateduration object from a date object to a new date object.
```

INPUTS

```
*this - date object
dd     - dateduration object
```

RESULT

```
d      - New date object
```

EXAMPLE

```
...  
d1 = d2 - dd;  
...
```

NOTES

None.

BUGS

The `dateduration` may not be longer (in days) than an integer value!

SEE ALSO

`operator+()`, `operator+=()`, `operator-=()`, `date()`, `dateduration/getindays()`,
`Date/HeisDiffDate()`

1.31 DateClass/date/operator-()

`DateClass/date/operator-()`

NAME

`operator-()` -- Sub operator for date object and weekday (V33)

SYNOPSIS

```
d = (*this) - wd;  
  
date operator-(const Weekdays wd) const;
```

FUNCTION

Gives you the last date, that is the specified weekday (could be the same date!).

INPUTS

```
*this - date object  
wd    - Weekday
```

RESULT

d - New date object

EXAMPLE

```
...  
d1 = d2 - wd;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator+=()`, `operator-=()`, `date()`,
`Date/HeisDaysBeforeWeekday()`, `Date/HeisDiffDate()`

1.32 DateClass/date/operator--()

`DateClass/date/operator--()`

NAME

`operator--()` -- Decrement operator for date object (V33)

SYNOPSIS

```
d--;  
  
date operator--();
```

FUNCTION

Decrements the date object by one day.

INPUTS

*this - date object

RESULT

*this - Modified date object

EXAMPLE

```
...  
dd--;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator++()`, `operator++(int)`, `operator--(int)`, `Date/HeisDiffDate()`

1.33 DateClass/date/operator--(int)

DateClass/date/operator--(int)

NAME

operator--(int) -- Decrement operator for date object (V33)

SYNOPSIS

```
--d;  
  
date operator--(int);
```

FUNCTION

Decrements the date object by one day.

INPUTS

*this - date object

RESULT

*this - Modified date object

EXAMPLE

```
...  
--d;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator++(), operator++(int), operator--(), Date/HeisDiffDate()

1.34 DateClass/date/operator-=()

DateClass/date/operator-=()

NAME

operator-=() -- Sub from operator for date and dateduration objs (V33)

SYNOPSIS

```
(*this) -= dd;  
  
const date &operator-=(const dateduration &dd);
```

FUNCTION

Subtract a dateduration object from a date object.

INPUTS

*this - date object
dd - dateduration object

RESULT

*this - Modified date object

EXAMPLE

```
...  
d -= dd;  
...
```

NOTES

None.

BUGS

The dateduration may not be longer (in days) than an integer value!

SEE ALSO

operator+(), operator-(), operator+=(), dateduration/getindays(),
Date/HeisDiffDate()

1.35 DateClass/date/operator-=()

DateClass/date/operator-=()

NAME

operator-=() -- Sub from operator for date object and weekday (V33)

SYNOPSIS

```
(*this) -= wd;  
  
const date &operator-=(const Weekdays wd);
```

FUNCTION

Corrects the date object to the last date that is the specified weekday (could be the same date!).

INPUTS

*this - date object
wd - Weekday

RESULT

*this - Modified date object

EXAMPLE

```
...  
d -= wd;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), Date/HeisDaysBeforeWeekday(),
Date/HeisDiffDate()

1.36 DateClass/date/operator<()

DateClass/date/operator<()

NAME

operator<() -- Lower than operator for date objects (V33)

SYNOPSIS

```
compare = ((*this) < d);  
  
bool operator<(const date &d) const;
```

FUNCTION

Compares if date object 1 is lower than date object 2.

INPUTS

*this - date object
d - date object

RESULT

compare - Boolean result of the lower than compare operation

EXAMPLE

```
...
if (d1 < d2)
    printf("<\n");
else
    printf(">=\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator>(), operator<=(), operator>=(),
Date/Compare2Dates()

1.37 DateClass/date/operator<()

DateClass/date/operator<()

NAME

operator<() -- Lower than operator for date object and weekday (V33)

SYNOPSIS

```
compare = ((*this) < wd);

bool operator< (const Weekdays wd) const;
```

FUNCTION

Compares if the date object is lower than a weekday.

INPUTS

*this - date object
wd - Weekday

RESULT

compare - Boolean result of the lower than compare operation

EXAMPLE


```
...
if (d1 < wd)
    printf("<\n");
else
    printf(">=\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator==()`, `operator!=()`, `operator>()`, `operator<=()`, `operator>=()`,
`Date/HeisWeekday()`

1.38 DateClass/date/operator<=()

DateClass/date/operator<=()

NAME

`operator<=()` -- Lower or equal operator for date objects (V33)

SYNOPSIS

```
compare = ((*this) <= d);

bool operator<=(const date &d) const;
```

FUNCTION

Compares if date object 1 is lower or equal than date object 2.

INPUTS

```
*this - date object
d      - date object
```

RESULT

`compare` - Boolean result of the lower or equal compare operation

EXAMPLE

```
...
if (d1 <= d2)
    printf("<=\n");
else
    printf(">\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator==()`, `operator!=()`, `operator<()`, `operator>()`, `operator>=()`.
`Date/Compare2Dates()`

1.39 DateClass/date/operator<=()

`DateClass/date/operator<=()`

NAME

`operator<=()` -- Lower or equal operator for date obj. and weekday (V33)

SYNOPSIS

```
compare = ((*this) <= wd);  
  
bool operator<=(const Weekdays wd) const;
```

FUNCTION

Compares if the date object is lower or equal to a weekday.

INPUTS

`*this` - date object
`wd` - Weekday

RESULT

`compare` - Boolean result of the lower or equal compare operation

EXAMPLE

```
...  
if (d1 <= wd)  
    printf("<=\n");  
else  
    printf(">\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator==()`, `operator!=()`, `operator<()`, `operator>()`, `operator>=()`,
`Date/HeisWeekday()`

1.40 DateClass/date/operator==()

`DateClass/date/operator==()`

NAME

`operator==()` -- Equal operator for date objects (V33)

SYNOPSIS

```
compare = ((*this) == d);  
  
bool operator==(const date &d) const;
```

FUNCTION

Compares two date objects for equality.

INPUTS

`*this` - date object
`d` - date object

RESULT

`compare` - Boolean result of the equal compare operation

EXAMPLE

```
...  
if (d1 == d2)  
    printf("==\n");  
else  
    printf("!=\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator!=()`, `operator<()`, `operator>()`, `operator<=()`, `operator>=()`,

Date/Compre2Dates()

1.41 DateClass/date/operator==()

DateClass/date/operator==()

NAME

operator==() -- Equal operator for date object and weekday (V33)

SYNOPSIS

```
compare = ((*this) == wd);  
  
bool operator==(const Weekdays wd) const;
```

FUNCTION

Compares if the date object is equal to a weekday.

INPUTS

*this - date object
wd - Weekday

RESULT

compare - Boolean result of the equal compare operation

EXAMPLE

```
...  
if (d1 == wd)  
    printf("==\n");  
else  
    printf("!=\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator!=(), operator<(), operator>(), operator<=(), operator>=(),
Date/HeisWeekday()

1.42 DateClass/date/operator>()

DateClass/date/operator>()

NAME

operator>() -- Greater than operator for date objects (V33)

SYNOPSIS

```
compare = ((*this) > d);

bool operator>(const date &d) const;
```

FUNCTION

Compares if date object 1 is greater than date object 2.

INPUTS

```
*this - date object
d      - date object
```

RESULT

compare - Boolean result of the greater than compare operation

EXAMPLE

```
...
if (d1 > d2)
    printf(">\n");
else
    printf("<=\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator<(), operator<=(), operator>=(),
Date/Compare2Dates()

1.43 DateClass/date/operator>()

DateClass/date/operator>()

NAME

operator>() -- Greater than operator for date object and weekday (V33)

SYNOPSIS

```
compare = ((*this) > wd);

bool operator> (const Weekdays wd) const;
```

FUNCTION

Compares if the date objects is greater than a weekday.

INPUTS

*this - date object
wd - Weekday

RESULT

compare - Boolean result of the greater than compare operation

EXAMPLE

```
...
if (d1 > wd)
    printf(">\n");
else
    printf("<=\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator<(), operator<=(), operator>=(),
Date/HeisWeekday()

1.44 DateClass/date/operator>=()

DateClass/date/operator>=()

NAME

operator>=() -- Greater or equal operator for date objects (V33)

SYNOPSIS

```
compare = ((*this) >= d);
```

```
bool operator>=(const date &d) const;
```

FUNCTION

Compares if date object 1 is greater or equal than date object 2.

INPUTS

```
*this - date object  
d      - date object
```

RESULT

compare - Boolean result of the greater or equal compare operation

EXAMPLE

```
...  
if (d1 >= d2)  
    printf(">=\n");  
else  
    printf("<\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator==( ), operator!=( ), operator<( ), operator>( ), operator<=( ),  
Date/Compare2Dates()
```

1.45 DateClass/date/operator>=()

DateClass/date/operator>=()

NAME

operator>=() -- Greater or equal oper. for date obj. and weekday (V33)

SYNOPSIS

```
compare = ((*this) >= wd);  
  
bool operator>=(const Weekdays wd) const;
```

FUNCTION

Compares if the date object is greater or equal to a weekday.

INPUTS

*this - date object
wd - Weekday

RESULT

compare - Boolean result of the greater or equal compare operation

EXAMPLE

```
...  
if (d1 >= wd)  
    printf(">=\n");  
else  
    printf("<\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator<(), operator>(), operator<=(),
Date/HeisWeekday()

1.46 DateClass/dateduration

DateClass/dateduration

NAME

dateduration -- Class for date durations (V33)

FUNCTION

This class maps the concept of date durations.

NOTES

None.

BUGS

No known bugs.

SEE ALSO

timeduration, time, date

1.47 DateClass/dateduration/dateduration()

DateClass/dateduration/dateduration()

NAME

dateduration() -- Constructor of dateduration class (V33)

SYNOPSIS

```
dd = dateduration(days=0,months=0,years=0);

dateduration(unsigned long days=0, unsigned short months=0,
              unsigned short years=0);
```

FUNCTION

Constructor of the dateduration class.

INPUTS

```
days    - Days of a dateduration
months   - Months of a dateduration
years    - Years of a dateduration
```

RESULT

dd - Dateduration object

EXAMPLE

```
...
dateduration ddl(14,0,0);
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

1.48 DateClass/dateduration/getdays()

DateClass/dateduration/getdays()

NAME

getdays() -- Get days of dateduration object (V33)

SYNOPSIS

```
days = (*this).getdays();  
  
unsigned long getdays(void) const;
```

FUNCTION

Gets the number of days from the dateduration object.

INPUTS

*this - dateduration object

RESULT

days - Days of the dateduration object

EXAMPLE

```
...  
days = dd.getdays();  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

getmonths(), getyears(), getindays()

1.49 DateClass/dateduration/getindays()

DateClass/dateduration/getindays()

NAME

getindays() -- Get dateduration in days of dateduration object (V33)

SYNOPSIS

```
days = (*this).getindays();  
  
unsigned long getindays(void) const;
```

FUNCTION

Gets duration of the `dateduration` object in days.

INPUTS

*this - `dateduration` object

RESULT

days - Duration of the `dateduration` object in days

EXAMPLE

```
...
days = dd.getindays();
...
```

NOTES

A month will be calculated as 30 days, a year as 365 days.

BUGS

No known bugs.

SEE ALSO

`getdays()`, `getmonths()`, `getyears()`

1.50 DateClass/dateduration/getmonths()

`DateClass/dateduration/getmonths()`

NAME

`getmonths()` -- Get months of `dateduration` object (V33)

SYNOPSIS

```
months = (*this).getmonths();

unsigned short getmonths(void) const;
```

FUNCTION

Gets the number of months from the `dateduration` object.

INPUTS

*this - `dateduration` object

RESULT

months - Months of the dateduration object

EXAMPLE

```
...
months = dd.getmonths();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

getdays(), getyears(), getindays()

1.51 DateClass/dateduration/getyears()

DateClass/dateduration/getyears()

NAME

getyears() -- Get years of dateduration object (V33)

SYNOPSIS

```
years = (*this).getyears();

unsigned short getyears(void) const;
```

FUNCTION

Gets the number of years from the dateduration object.

INPUTS

*this - dateduration object

RESULT

years - Years of the dateduration object

EXAMPLE

```
...
years = dd.getyears();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`getdays()`, `getmonths()`, `getindays()`

1.52 DateClass/dateduration/operator!=()

DateClass/dateduration/operator!=()

NAME

`operator!=()` -- Unequal operator for dateduration objects (V33)

SYNOPSIS

```
compare = ((*this) != dd);  
  
bool operator!=(const dateduration &dd) const;
```

FUNCTION

Compares two dateduration objects for unequality.

INPUTS

```
*this - dateduration object  
dd    - dateduration object
```

RESULT

`compare` - Boolean result of the unequal compare operation

EXAMPLE

```
...  
if (ddl != dd2)  
    printf("!=\n");  
else  
    printf("==\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator==(), operator<(), operator>(), operator<=(), operator>=()`

1.53 DateClass/dateduration/operator%()

DateClass/dateduration/operator%()

NAME

`operator%()` -- Modulo operator for dateduration object (V33)

SYNOPSIS

```
ddf = (*this) % dd;
```

```
unsigned long operator%(const dateduration &dd) const;
```

FUNCTION

Modulo divide the dateduration object by another dateduration object to the modulo factor.

INPUTS

`*this` - dateduration object
`dd` - dateduration object

RESULT

`ddf` - Modulo factor

EXAMPLE

```
...  
ddf = dd1 % dd2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+(), operator-(), operator+=(), operator-=(), operator*(),
operator/(), operator*=(), operator/=()`

1.54 DateClass/dateduration/operator*()

DateClass/dateduration/operator*()

NAME

operator*() -- Multiply operator for dateduration object (V33)

SYNOPSIS

```
ddres = (*this) * ddm;
```

```
dateduration operator*(const unsigned short ddm) const;
```

FUNCTION

Multiply the dateduration object with a constant value to a new dateduration object.

INPUTS

*this - dateduration object
ddm - Multiplication factor

RESULT

ddres - New dateduration object

EXAMPLE

```
...  
ddres = dd * 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+ = (), operator- = (), operator/(),
operator* = (), operator/=(), dateduration()

1.55 DateClass/dateduration/operator*()

DateClass/dateduration/operator*()

NAME

operator*() -- Multiply operator for dateduration object (V33)

SYNOPSIS

```
ddres = (*this) * ddm;
```

```
dateduration operator*(const unsigned int ddm) const;
```

FUNCTION

Multiply the dateduration object with a constant value to a new dateduration object.

INPUTS

*this - dateduration object
ddm - Multiplication factor

RESULT

ddres - New dateduration object

EXAMPLE

```
...  
ddres = dd * 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator/(),
operator*=(), operator/=(), dateduration()

1.56 DateClass/dateduration/operator*()

DateClass/dateduration/operator*()

NAME

operator*() -- Multiply operator for dateduration object (V33)

SYNOPSIS

```
ddres = (*this) * ddm;
```

```
dateduration operator*(const unsigned long ddm) const;
```

FUNCTION

Multiply the dateduration object with a constant value to a new

dateduration object.

INPUTS

*this - dateduration object
ddm - Multiplication factor

RESULT

ddres - New dateduration object

EXAMPLE

```
...  
ddres = dd * 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator/(),
operator*=(), operator/=(), dateduration()

1.57 DateClass/dateduration/operator*=()

DateClass/dateduration/operator*=()

NAME

operator*=() -- Multiply with operator for dateduration object (V33)

SYNOPSIS

```
(*this) *= ddm;  
  
const dateduration &operator*=(const unsigned short ddm);
```

FUNCTION

Multiply the dateduration object with a constant value.

INPUTS

*this - dateduration object
ddm - Multiplication factor

RESULT

*this - Modified dateduration object

EXAMPLE

```
...  
dd *= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator*(),
operator/(), operator/=()

1.58 DateClass/dateduration/operator*=()

DateClass/dateduration/operator*=()

NAME

operator*=() -- Multiply with operator for dateduration object (V33)

SYNOPSIS

```
(*this) *= ddm;  
  
const dateduration &operator*=(const unsigned int ddm);
```

FUNCTION

Multiply the dateduration object with a constant value.

INPUTS

*this - dateduration object
ddm - Multiplication factor

RESULT

*this - Modified dateduration object

EXAMPLE

```
...  
dd *= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator+=()`, `operator-=()`, `operator*()`,
`operator/()`, `operator/=()`

1.59 DateClass/dateduration/operator*=()

DateClass/dateduration/operator*=()

NAME

`operator*=()` -- Multiply with operator for dateduration object (V33)

SYNOPSIS

```
(*this) *= ddm;  
  
const dateduration &operator*=(const unsigned long ddm);
```

FUNCTION

Multiply the dateduration object with a constant value.

INPUTS

`*this` - dateduration object
`ddm` - Multiplication factor

RESULT

`*this` - Modified dateduration object

EXAMPLE

```
...  
dd *= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator+(), operator-(), operator+==(), operator-==(), operator*(),  
operator/(), operator/=()
```

1.60 DateClass/dateduration/operator+()

DateClass/dateduration/operator+()

NAME

operator+() -- Add operator for dateduration objects (V33)

SYNOPSIS

```
ddres = (*this) + dd;  
  
dateduration operator+(const dateduration &dd) const;
```

FUNCTION

Add two dateduration objects to a new dateduration object.

INPUTS

```
*this - dateduration object  
dd    - dateduration object
```

RESULT

ddres - new dateduration object

EXAMPLE

```
...  
ddres = dd1 + dd2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator-(), operator+==(), operator-==(), operator*(), operator/(),  
operator*=(), operator/=(), dateduration()
```

1.61 DateClass/dateduration/operator++()

DateClass/dateduration/operator++()

NAME

operator++() -- Increment operator for dateduration object (V33)

SYNOPSIS

```
dd++;

dateduration operator++();
```

FUNCTION

Increments the dateduration object by one day.

INPUTS

*this - dateduration object

RESULT

*this - modified dateduration object

EXAMPLE

```
...
dd++;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator++(int), operator--(), operator--(int)

1.62 DateClass/dateduration/operator++(int)

DateClass/dateduration/operator++(int)

NAME

operator++(int) -- Increment operator for dateduration object (V33)

SYNOPSIS

```
++dd;

dateduration operator++(int);
```

FUNCTION

Increments the `dateduration` object by one day.

INPUTS

*this - `dateduration` object

RESULT

*this - modified `dateduration` object

EXAMPLE

```
...
++dd;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator++()`, `operator--()`, `operator--(int)`

1.63 DateClass/dateduration/operator+=()

DateClass/dateduration/operator+=()

NAME

`operator+=()` -- Add to operator for `dateduration` objects (V33)

SYNOPSIS

```
(*this) += dd;

const dateduration &operator+=(const dateduration &dd);
```

FUNCTION

Add `dateduration` object 2 to `dateduration` object 1.

INPUTS

*this - `dateduration` object
dd - `dateduration` object

RESULT

*this - modified dateduration object

EXAMPLE

```
...  
dd1 += dd2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator-=(), operator*(), operator/(),
operator*=(), operator/=()

1.64 DateClass/dateduration/operator-()

DateClass/dateduration/operator-()

NAME

operator-() -- Sub operator for dateduration objects (V33)

SYNOPSIS

```
ddres = (*this) - dd;
```

```
dateduration operator-(const dateduration &dd) const;
```

FUNCTION

Subtract dateduration object 2 from dateduration object 1 to a new dateduration object.

If object 2 is greater than object 1 you get back an object with a duration of 0!

INPUTS

*this - dateduration object
dd - dateduration object

RESULT

ddres - new dateduration object

EXAMPLE

```
...
```

```
ddres = dd1 - dd2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator+(), operator+=(), operator-=(), operator*(), operator/(),  
operator*=(), operator/=(), dateduration()
```

1.65 DateClass/dateduration/operator--()

DateClass/dateduration/operator--()

NAME

operator--() -- Decrement operator for dateduration object (V33)

SYNOPSIS

```
dd--;  
  
dateduration operator--();
```

FUNCTION

Decrements the dateduration object by one day.

INPUTS

*this - dateduration object

RESULT

*this - modified dateduration object

EXAMPLE

```
...  
dd--;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator++()`, `operator++(int)`, `operator--(int)`

1.66 DateClass/dateduration/operator--(int)

DateClass/dateduration/operator--(int)

NAME

`operator--()` -- Decrement operator for dateduration object (V33)

SYNOPSIS

```
--dd;  
  
dateduration operator--(int);
```

FUNCTION

Decrements the dateduration object by one day.

INPUTS

*this - dateduration object

RESULT

*this - modified dateduration object

EXAMPLE

```
...  
--dd;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator++()`, `operator++(int)`, `operator--()`

1.67 DateClass/dateduration/operator-=()

DateClass/dateduration/operator-=()

NAME

operator-=() -- Sub from operator for dateduration objects (V33)

SYNOPSIS

```
(*this) -= dd;
```

```
const dateduration &operator-=(const dateduration &dd);
```

FUNCTION

Subtract dateduration object 2 from dateduration object 1.
If object 2 is greater than object 1 you get back an object with a duration of 0!

INPUTS

*this - dateduration object
dd - dateduration object

RESULT

*this - modified dateduration object

EXAMPLE

```
...  
ddl -= dd2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator*(), operator/(),
operator*=(), operator/=()

1.68 DateClass/dateduration/operator/()

DateClass/dateduration/operator/()

NAME

operator/() -- Div operator for dateduration object (V33)

SYNOPSIS

```
ddres = (*this) / ddd;

dateduration operator/(const unsigned short ddd) const;
```

FUNCTION

Devide the dateduration object by a constant value to a new dateduration object.
If ddd is 0 you will get back an object with a duration of 0!

INPUTS

*this - dateduration object
ddd - Division factor

RESULT

ddres - New dateduration object

EXAMPLE

```
...
ddres = dd / 2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator*(),
operator*=(), operator/=(), dateduration()

1.69 DateClass/dateduration/operator/()

DateClass/dateduration/operator/()

NAME

operator/() -- Div operator for dateduration object (V33)

SYNOPSIS

```
ddres = (*this) / ddd;

dateduration operator/(const unsigned int ddd) const;
```

FUNCTION

Devide the dateduration object by a constant value to a new dateduration object.
If ddd is 0 you will get back an object with a duration of 0!

INPUTS

*this - dateduration object
ddd - Division factor

RESULT

ddres - New dateduration object

EXAMPLE

```
...  
ddres = dd / 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator*(),
operator*=(), operator/=(), dateduration()

1.70 DateClass/dateduration/operator/()

DateClass/dateduration/operator/()

NAME

operator/() -- Div operator for dateduration object (V33)

SYNOPSIS

```
ddres = (*this) / ddd;  
  
dateduration operator/(const unsigned long ddd) const;
```

FUNCTION

Devide the dateduration object by a constant value to a new dateduration object.
If ddd is 0 you will get back an object with a duration of 0!

INPUTS

```
*this - dateduration object  
ddd   - Division factor
```

RESULT

ddres - New dateduration object

EXAMPLE

```
...  
ddres = dd / 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator+(), operator-(), operator+=(), operator-=(), operator*(),  
operator*=(), operator/=( ), dateduration()
```

1.71 DateClass/dateduration/operator/()

DateClass/dateduration/operator/()

NAME

operator/() -- Div operator for dateduration object (V33)

SYNOPSIS

```
ddf = (*this) / dd;
```

```
unsigned long operator/(const dateduration &dd) const;
```

FUNCTION

Devide the dateduration object by another dateduration object to the devision factor.

INPUTS

```
*this - dateduration object  
dd     - dateduration object
```

RESULT

ddf - Division factor

EXAMPLE

```
...  
ddf = dd1 / dd2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator+=()`, `operator-=()`, `operator*()`,
`operator*=()`, `operator/=(())`, `operator%()`

1.72 DateClass/dateduration/operator/=(())

DateClass/dateduration/operator/=(())

NAME

`operator/=(())` -- Div through operator for dateduration object (V33)

SYNOPSIS

```
(*this) /= ddd;  
  
const dateduration &operator/=(const unsigned short ddd);
```

FUNCTION

Devide the dateduration object by a constant value.
If ddd is 0 you will get back an object with a duration of 0!

INPUTS

`*this` - dateduration object
`ddd` - Division factor

RESULT

`*this` - Modified dateduration object

EXAMPLE

```
...  
dd /= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator+=()`, `operator-=()`, `operator*()`,
`operator/()`, `operator*=()`

1.73 DateClass/dateduration/operator/=()

DateClass/dateduration/operator/=()

NAME

`operator/=()` -- Div through operator for dateduration object (V33)

SYNOPSIS

```
(*this) /= ddd;
```

```
const dateduration &operator/=(const unsigned int ddd);
```

FUNCTION

Devide the dateduration object by a constant value.
If ddd is 0 you will get back an object with a duration of 0!

INPUTS

`*this` - dateduration object
`ddd` - Division factor

RESULT

`*this` - Modified dateduration object

EXAMPLE

```
...  
dd /= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator+=()`, `operator-=()`, `operator*()`,

```
operator/(),operator*=(())
```

1.74 DateClass/dateduration/operator/=()

DateClass/dateduration/operator/=()

NAME

operator/=() -- Div through operator for dateduration object (V33)

SYNOPSIS

```
(*this) /= ddd;  
  
const dateduration &operator/=(const unsigned long ddd);
```

FUNCTION

Devide the dateduration object by a constant value.
If ddd is 0 you will get back an object with a duration of 0!

INPUTS

*this - dateduration object
ddd - Division factor

RESULT

*this - Modified dateduration object

EXAMPLE

```
...  
dd /= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(),operator-(),operator+=(),operator-=(),operator*(),
operator/(),operator*=(())

1.75 DateClass/dateduration/operator<=()

DateClass/dateduration/operator<=()

NAME

operator<=() -- Lower or equal operator for dateduration objects (V33)

SYNOPSIS

```
compare = ((*this) <= dd);

bool operator<=(const dateduration &dd) const;
```

FUNCTION

Compares if dateduration object 1 is lower or equal than dateduration object 2.

INPUTS

*this - dateduration object
dd - dateduration object

RESULT

compare - Boolean result of the lower or equal compare operation

EXAMPLE

```
...
if (ddl <= dd2)
    printf("<=\n");
else
    printf(">\n");
...
```

NOTES

A month will be calculated as 30 days, a year as 365 days.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator<(), operator>(), operator>=()

1.76 DateClass/dateduration/operator==()

DateClass/dateduration/operator==()

NAME

operator==() -- Equal operator for dateduration objects (V33)

SYNOPSIS

```
compare = ((*this) == dd);

bool operator==(const dateduration &dd) const;
```

FUNCTION

Compares two dateduration objects for equality.

INPUTS

*this - dateduration object
dd - dateduration object

RESULT

compare - Boolean result of the equal compare operation

EXAMPLE

```
...
if (ddl == dd2)
    printf("==\n");
else
    printf("!=\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator!=(), operator<(), operator>(), operator<=(), operator>=()

1.77 DateClass/dateduration/operator>()

DateClass/dateduration/operator>()

NAME

operator>() -- Greater than operator for dateduration objects (V33)

SYNOPSIS

```
compare = ((*this) > dd);

bool operator>(const dateduration &dd) const;
```

FUNCTION

Compares if dateduration object 1 is greater than dateduration object 2.

INPUTS

*this - dateduration object
dd - dateduration object

RESULT

compare - Boolean result of the greater than compare operation

EXAMPLE

```
...  
if (dd1 > dd2)  
    printf(">\n");  
else  
    printf("<=\n");  
...
```

NOTES

A month will be calculated as 30 days, a year as 365 days.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator<(), operator<=(), operator>=()

1.78 DateClass/dateduration/operator>=()

DateClass/dateduration/operator>=()

NAME

operator>=() -- Greater or equal operator for dateduration objs (V33)

SYNOPSIS

```
compare = ((*this) >= dd);  
  
bool operator>=(const dateduration &dd) const;
```

FUNCTION

Compares if dateduration object 1 is greater or equal than dateduration object 2.

INPUTS

*this - dateduration object
dd - dateduration object

RESULT

compare - Boolean result of the greater or equal compare operation

EXAMPLE

```
...  
if (ddl >= dd2)  
    printf(">=\n");  
else  
    printf("<\n");  
...
```

NOTES

A month will be calculated as 30 days, a year as 365 days.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator<(), operator>(), operator<=()

1.79 DateClass/dateeduration/operator<()

DateClass/dateeduration/operator<()

NAME

operator<() -- Lower than operator for dateduration objects (V33)

SYNOPSIS

```
compare = ((*this) < dd);  
  
bool operator<(const dateduration &dd) const;
```

FUNCTION

Compares if dateduration object 1 is lower than dateduration object 2.

INPUTS

*this - dateduration object
dd - dateduration object

RESULT

compare - Boolean result of the lower than compare operation

EXAMPLE

```
...
if (dd1 < dd2)
    printf("<\n");
else
    printf(">=\n");
...
```

NOTES

A month will be calculated as 30 days, a year as 365 days.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator>(), operator<=(), operator>=()

1.80 DateClass/time

DateClass/time

NAME

time -- Class for times. (V33)

FUNCTION

This class maps 24h times.

NOTES

None.

BUGS

No known bugs.

SEE ALSO

timeduration, dateduration, date

1.81 DateClass/time/gethour()

DateClass/time/gethour()

NAME

gethour() -- Get hour of time object (V33)

SYNOPSIS

```
hour = (*this).gethour();

unsigned short gethour(void) const;
```

FUNCTION

Get the hour from the time object.

INPUTS

*this - time object

RESULT

hour - Hour of the time object

EXAMPLE

```
...
hour = t.gethour();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

getmin(), getsec(), getzone(), getjd()

1.82 DateClass/time/getjd()

DateClass/time/getjd()

NAME

getjd() -- Get time in JD format from time object (V33)

SYNOPSIS

```
jd = (*this).getjd();

float getjd(void) const;
```

FUNCTION

Get the time in JD format from the time object.

INPUTS

*this - time object

RESULT

jd - Time of the time object in JD format

EXAMPLE

```
...  
jd = t.getjd();  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

gethour(), getmin(), getsec(), getzone(), Date/TimeToJD()

1.83 DateClass/time/getmin()

DateClass/time/getmin()

NAME

getmin() -- Get min of time object (V33)

SYNOPSIS

```
min = (*this).getmin();  
  
unsigned short getmin(void) const;
```

FUNCTION

Get the minute from the time object.

INPUTS

*this - time object

RESULT

min - Minute of the time object

EXAMPLE

```
...
min = t.getmin();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`gethour()`, `getsec()`, `getzone()`, `getjd()`

1.84 DateClass/time/getsec()

DateClass/time/getsec()

NAME

`getsec()` -- Get sec of time object (V33)

SYNOPSIS

```
sec = (*this).getsec();

unsigned short getsec(void) const;
```

FUNCTION

Get the second from the time object.

INPUTS

*this - time object

RESULT

sec - Second of the time object

EXAMPLE

```
...
sec = t.getsec();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`gethour()`, `getmin()`, `getzone()`, `getjd()`

1.85 DateClass/time/getzone()

DateClass/time/getzone()

NAME

`getzone()` -- Get timezone of time object (V33)

SYNOPSIS

```
zone = (*this).getzone();  
  
short getzone(void) const;
```

FUNCTION

Get the timezone from the time object.

INPUTS

*this - time object

RESULT

zone - Timezone of the time object

EXAMPLE

```
...  
zone = t.getzone();  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`gethour()`, `getmin()`, `getsec()`, `getjd()`

1.86 DateClass/time/operator!=()

DateClass/time/operator!=()

NAME

operator!=() -- Unequal operator for time objects (V33)

SYNOPSIS

```
compare = ((*this) != t);

bool operator!=(time t) const;
```

FUNCTION

Compares two time objects for unequality.

INPUTS

```
*this - time object
t      - time object
```

RESULT

compare - Boolean result of the unequal compare operation

EXAMPLE

```
...
if (t1 != t2)
    printf("!=\n");
else
    printf("==\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator==(), operator<(), operator>(), operator<=(), operator>=(),
operator[](), Date/Compare2Times()

1.87 DateClass/time/operator()()

DateClass/time/operator()()

NAME

operator() () -- Calc. to new timezone operator for time object (V33)

SYNOPSIS

```
t = (*this) (zone);

time operator() (const short zone) const;
```

FUNCTION

Change the timezone of a time object to a new time object with another timezone

INPUTS

*this - time object
zone - Add this value to the actual timezone

RESULT

t - New time object

EXAMPLE

```
...
t1 = t2(2);
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator<<(), operator>>(), operator<=(), operator>=(), operator[](),
time(), Date/DiffTime()

1.88 DateClass/time/operator+()

DateClass/time/operator+()

NAME

operator+() -- Add operator for time and timeduration objects (V33)

SYNOPSIS

```
t = (*this) + td;
```

```
time operator+(const timeduration &td) const;
```

FUNCTION

Add a time object and a timeduration object to a new time object.

INPUTS

```
*this - time object  
td    - timeduration object
```

RESULT

t - New time object

EXAMPLE

```
...  
t1 = t2 + td;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator-()`, `operator+=()`, `operator-=()`, `time()`, `timeduration/getinsecs()`,
`Date/DiffTime()`

1.89 DateClass/time/operator+=()

DateClass/time/operator+=()

NAME

`operator+=()` -- Add to operator for time and timeduration objects (V33)

SYNOPSIS

```
(*this) += td;  
  
const time &operator+=(const timeduration &td);
```

FUNCTION

Add a timeduration object to a time object.

INPUTS

```
*this - time object
```

td - timeduration object

RESULT

*this - Modified time object

EXAMPLE

```
...  
t += td;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator-=(), timeduration/getinsecs(),
Date/DiffTime()

1.90 DateClass/time/operator-()

DateClass/time/operator-()

NAME

operator-() -- Sub operator for time objects (V33)

SYNOPSIS

```
td = (*this) - t;  
  
timeduration operator-(time t) const;
```

FUNCTION

Subtract time object 2 from time object 1 to a new timeduration object.

INPUTS

*this - time object
t - time object

RESULT

td - New timeduration object

EXAMPLE

```
...
td = t1 - t2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `timeduration()`, `operator[]()`, `Date/TimeDiff()`,
`Date/SecToTime()`

1.91 DateClass/time/operator-()

DateClass/time/operator-()

NAME

`operator-()` -- Sub operator for time and timeduration objects (V33)

SYNOPSIS

```
t = (*this) - td;

time operator-(const timeduration &td) const;
```

FUNCTION

Subtract a timeduration object from a time object to a new time object.

INPUTS

```
*this - time object
td    - timeduration object
```

RESULT

t - New time object

EXAMPLE

```
...
t1 = t2 - td;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator+=()`, `operator-=()`, `time()`, `timeduration/getinsecs()`,
`Date/DiffTime()`,

1.92 DateClass/time/operator-=()

DateClass/time/operator-=()

NAME

`operator-=()` -- Sub from operator for time and timeduration objs (V33)

SYNOPSIS

```
(*this) -= td;  
  
const time &operator-=(const timeduration &td);
```

FUNCTION

Subtract a timeduration object from a time object.

INPUTS

`*this` - time object
`td` - timeduration object

RESULT

`*this` - Modified time object

EXAMPLE

```
...  
t -= td;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator+=()`, `timeduration/getinsecs()`,
`Date/DiffTime()`

1.93 DateClass/time/operator<()

DateClass/time/operator<()

NAME

operator<() -- Lower than operator for time objects (V33)

SYNOPSIS

```
compare = ((*this) < t);  
  
bool operator<(time t) const;
```

FUNCTION

Compares if time object 1 is lower than time object 2.

INPUTS

```
*this - time object  
t      - time object
```

RESULT

compare - Boolean result of the lower than compare operation

EXAMPLE

```
...  
if (t1 < t2)  
    printf("<\n");  
else  
    printf(">=\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator==( ), operator!=( ), operator>( ), operator<=( ), operator>=( ),  
operator[] ( ), Date/Compare2Times ( )
```

1.94 DateClass/time/operator<<()

DateClass/time/operator<<()

NAME

operator<<() -- Decrement timezone operator for time object (V33)

SYNOPSIS

```
t = (*this) << subzone;

time operator<<(const unsigned short subzone) const;
```

FUNCTION

Decrement the timezone of a time object to a new time object.

INPUTS

*this - time object
subzone - Subtract this value from the actual timezone

RESULT

t - New time object

EXAMPLE

```
...
t1 = t2 << 2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator>>(), operator<=(), operator>=(), operator[](), operator()(),
time(), Date/DiffTime()

1.95 DateClass/time/operator<=()

DateClass/time/operator<=()

NAME

operator<=() -- Decrement timezone operator for time object (V33)

SYNOPSIS

```
(*this) <= subzone;
```

```
const time &operator<=(const unsigned short subzone);
```

FUNCTION

Decrement the timezone of a time object.

INPUTS

*this - time object
subzone - Subtract this value from the actual timezone

RESULT

*this - Modified time object

EXAMPLE

```
...  
t <= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator<<(), operator>>(), operator>=(), operator[](), operator()(),
Date/DiffTime()

1.96 DateClass/time/operator<=()

DateClass/time/operator<=()

NAME

operator<=() -- Lower or equal operator for time objects (V33)

SYNOPSIS

```
compare = ((*this) <= t);  
  
bool operator<=(time t) const;
```

FUNCTION

Compares if time object 1 is lower or equal than time object 2.

INPUTS

```
*this - time object  
t      - time object
```

RESULT

compare - Boolean result of the lower or equal compare operation

EXAMPLE

```
...  
if (t1 <= t2)  
    printf("<=\n");  
else  
    printf(">\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator<(), operator>(), operator>=(),
operator[] (), Date/Compare2Times ()

1.97 DateClass/time/operator==()

DateClass/time/operator==()

NAME

operator==() -- Equal operator for time objects (V33)

SYNOPSIS

```
compare = ((*this) == t);  
  
bool operator==(time t) const;
```

FUNCTION

Compares two time objects for equality.

INPUTS

```
*this - time object  
t      - time object
```

RESULT

compare - Boolean result of the equal compare operation

EXAMPLE

```
...
if (t1 == t2)
    printf("==\n");
else
    printf("!=\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator!=()`, `operator<()`, `operator>()`, `operator<=()`, `operator>=()`,
`operator[]()`, `Date/Compare2Times()`

1.98 DateClass/time/operator>()

DateClass/time/operator>()

NAME

`operator>()` -- Greater than operator for time objects (V33)

SYNOPSIS

```
compare = ((*this) > t);

bool operator>(time t) const;
```

FUNCTION

Compares if time object 1 is greater than time object 2.

INPUTS

```
*this - time object
t      - time object
```

RESULT

compare - Boolean result of the greater than compare operation

EXAMPLE

```
...
if (t1 > t2)
    printf(">\n");
```

```
else
    printf("<=\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator==( ), operator!=( ), operator<( ), operator<=( ), operator>=( ),
operator[]( ), Date/Compare2Times()
```

1.99 DateClass/time/operator>=()

DateClass/time/operator>=()

NAME

operator>=() -- Greater or equal operator for time objects (V33)

SYNOPSIS

```
compare = ((*this) >= t);

bool operator>=(time t) const;
```

FUNCTION

Compares if time object 1 is greater or equal than time object 2.

INPUTS

```
*this - time object
t      - time object
```

RESULT

compare - Boolean result of the greater or equal compare operation

EXAMPLE

```
...
if (t1 >= t2)
    printf(">=\n");
else
    printf("<\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator==()`, `operator!=()`, `operator<()`, `operator>()`, `operator<=()`,
`operator[]()`, `Date/Compare2Times()`

1.100 DateClass/time/operator>>()

`DateClass/time/operator>>()`

NAME

`operator>>()` -- Increment timezone operator for time object (V33)

SYNOPSIS

```
t = (*this) >> addzone;  
  
time operator>>(const unsigned short addzone) const;
```

FUNCTION

Increment the timezone of a time object to a new time object.

INPUTS

`*this` - time object
`addzone` - Add this value to the actual timezone

RESULT

`t` - New time object

EXAMPLE

```
...  
t1 = t2 >> 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator<<()`, `operator<=()`, `operator>>=()`, `operator[]()`, `operator()()`,

```
time(),Date/DiffTime()
```

1.101 DateClass/time/operator>>=()

DateClass/time/operator>>=()

NAME

operator>>=() -- Increment timezone operator for time object (V33)

SYNOPSIS

```
(*this) >>= addzone;  
  
const time &operator>>=(const unsigned short addzone);
```

FUNCTION

Increment the timezone of a time object.

INPUTS

*this - time object
addzone - Add this value to the actual timezone

RESULT

*this - Modified time object

EXAMPLE

```
...  
t >>= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator<<(),operator>>(),operator<<=(),operator[](),operator()(),
Date/DiffTime()

1.102 DateClass/time/operator[]()

DateClass/time/operator[]()

NAME

operator[]() -- Change to timezone operator for time object (V33)

SYNOPSIS

```
(*this) = (*this)[zone];  
  
const time &operator[](const short zone);
```

FUNCTION

Change the timezone of a time object to another timezone.

INPUTS

*this - time object
zone - Add this value to the actual timezone

RESULT

*this - Modified time object

EXAMPLE

```
...  
t[2];  
t1 = t2[2];  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator<<(), operator>>(), operator<=(), operator>=(), operator()(),
Date/DiffTime()

1.103 DateClass/time/time()

DateClass/time/time()

NAME

time() -- Constructor of time class (V33)

SYNOPSIS

```
t = time(jd,zone=0);  
  
time(float jd, short zone = 0);
```

FUNCTION

Constructor of the time class using the JD format.

INPUTS

jd - Time in JD format
zone - Timezone of the time

RESULT

t - Time object

EXAMPLE

```
...  
time t(0.5,2);  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

time(), Date/JDToTime()

1.104 DateClass/time/time()

DateClass/time/time()

NAME

time() -- Constructor of time class (V33)

SYNOPSIS

```
t = time(jd,zone=0);  
  
time(double jd, short zone = 0);
```

FUNCTION

Constructor of the time class using the JD format.

INPUTS

jd - Time in JD format
zone - Timezone of the time

RESULT

t - Time object

EXAMPLE

```
...  
time t(0.5,2);  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

time(), Date/JDToTime()

1.105 DateClass/time/time()

DateClass/time/time()

NAME

time() -- Constructor of time class (V33)

SYNOPSIS

```
t = time(hour=0,min=0,sec=0,zone=0);
```

```
time(unsigned short hour=0, unsigned short min=0, unsigned short sec=0,  
short zone=0);
```

FUNCTION

Constructor of the time class.

INPUTS

hour - Hour of time
min - Min of time
sec - Sec of time
zone - Timezone of time

RESULT

t - Time object

EXAMPLE

```
...  
time t(12,1,58,2);  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

time()

1.106 DateClass/timeduration

DateClass/timeduration

NAME

timeduration -- Class for time durations (V33)

FUNCTION

This class maps the concept of time durations.

NOTES

None.

BUGS

No known bugs.

SEE ALSO

dateduration,time,date

1.107 DateClass/timeduration/gethours()

DateClass/timeduration/gethours()

NAME

gethours() -- Get hours of timeduration object (V33)

SYNOPSIS

```
hours = (*this).gethours();  
  
unsigned short gethours(void) const;
```

FUNCTION

Gets the number of hours from the timeduration object.

INPUTS

*this - timeduration object

RESULT

hours - Hours of the timeduration object

EXAMPLE

```
...  
hours = td.gethours();  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

getmins(), getsecs(), getinsecs()

1.108 DateClass/timeduration/getinsecs()

DateClass/timeduration/getinsecs()

NAME

getinsecs() -- Get timeduration in secs of timeduration object (V33)

SYNOPSIS

```
secs = (*this).getinsecs();  
  
unsigned long getinsecs(void) const;
```

FUNCTION

Gets duration of the timeduration object in seconds.

INPUTS

*this - timeduration object

RESULT

secs - Timeduration of the timeduration object in seconds.

EXAMPLE

```
...
secs = td.getinsecs();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

gethours(), getmins(), getsecs(), Date/TimeToSec()

1.109 DateClass/timeduration/getmins()

DateClass/timeduration/getmins()

NAME

getmins() -- Get mins of timeduration object (V33)

SYNOPSIS

```
mins = (*this).getmins();

unsigned short getmins(void) const;
```

FUNCTION

Gets the number of minutes from the timeduration object.

INPUTS

*this - timeduration object

RESULT

mins - Minutes of the timeduration object

EXAMPLE

```
...
mins = td.getmins();
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`gethours()`, `getsecs()`, `getinsecs()`

1.110 DateClass/timeduration/getsecs()

DateClass/timeduration/getsecs()

NAME

`getsecs()` -- Get secs of timeduration object (V33)

SYNOPSIS

```
secs = (*this).getsecs();  
  
unsigned short getsecs(void) const;
```

FUNCTION

Gets the number of seconds from the timeduration object.

INPUTS

*this - timeduration object

RESULT

secs - Seconds of the timeduration object

EXAMPLE

```
...  
secs = td.getsecs();  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`gethours()`, `getmins()`, `getinsecs()`

1.111 DateClass/timeduration/operator!=()

DateClass/timeduration/operator!=()

NAME

operator!=() -- Unequal operator for timeduration objects (V33)

SYNOPSIS

```
compare = ((*this) != td);

bool operator!=(const timeduration &td) const;
```

FUNCTION

Compares two timeduration objects for unequality.

INPUTS

*this - timeduration object
td - timeduration object

RESULT

compare - Boolean result of the unequal compare operation

EXAMPLE

```
...
if (td1 != td2)
    printf("!=\n");
else
    printf("==\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator==(), operator<(), operator>(), operator<=(), operator>=(),
Date/Compare2Times()

1.112 DateClass/timeduration/operator%()

DateClass/timeduration/operator%()

NAME

operator%() -- Modulo operator for timeduration object (V33)

SYNOPSIS

```
tdf = (*this) % td;
```

```
unsigned long operator%(const timeduration &td) const;
```

FUNCTION

Modulo devide the timeduration object by another timeduration object to the modulo factor.

INPUTS

*this - timeduration object

td - timeduration object

RESULT

tdf - Modulo factor

EXAMPLE

```
...  
tdf = td1 % td2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator*(),
operator/(), operator*=(), operator/=(), Date/TimeToSec()

1.113 DateClass/timeduration/operator*()

DateClass/timeduration/operator*()

NAME

operator*() -- Multiply operator for timeduration object (V33)

SYNOPSIS

```
tdres = (*this) * tdm;

timeduration operator*(const unsigned short tdm) const;
```

FUNCTION

Multiply the timeduration object with a constant value to a new timeduration object.

INPUTS

*this - timeduration object
tdm - Multiplication factor

RESULT

tdres - new timeduration object

EXAMPLE

```
...
tdres = td * 2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator/(),
operator*=(), operator/=(), timeduration(), Date/TimeToSec(),
Date/SecToTime()

1.114 DateClass/timeduration/operator*()

DateClass/timeduration/operator*()

NAME

operator*() -- Multiply operator for timeduration object (V33)

SYNOPSIS

```
tdres = (*this) * tdm;

timeduration operator*(const unsigned int tdm) const;
```

FUNCTION

Multiply the timeduration object with a constant value to a new timeduration object.

INPUTS

*this - timeduration object
tdm - Multiplication factor

RESULT

tdres - new timeduration object

EXAMPLE

```
...  
tdres = td * 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator/(),
operator*(), operator/=(), timeduration(), Date/TimeToSec(),
Date/SecToTime()

1.115 DateClass/timeduration/operator*()

DateClass/timeduration/operator*()

NAME

operator*() -- Multiply operator for timeduration object (V33)

SYNOPSIS

```
tdres = (*this) * tdm;  
  
timeduration operator*(const unsigned long tdm) const;
```

FUNCTION

Multiply the timeduration object with a constant value to a new timeduration object.

INPUTS

*this - timeduration object

tdm - Multiplication factor

RESULT

tdres - new timeduration object

EXAMPLE

```
...  
tdres = td * 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator/(),
operator*(), operator/=(), timeduration(), Date/TimeToSec(),
Date/SecToTime()

1.116 DateClass/timeduration/operator*=()

DateClass/timeduration/operator*=()

NAME

operator*=() -- Multiply with operator for timeduration object (V33)

SYNOPSIS

```
(*this) *= tdm;  
  
const timeduration &operator*=(const unsigned short tdm);
```

FUNCTION

Multiply the timeduration object with a constant value.

INPUTS

*this - timeduration object
tdm - Multiplication factor

RESULT

*this - Modified timeduration object

EXAMPLE

```
...
td *= 2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator+=()`, `operator-=()`, `operator*()`,
`operator/()`, `operator/=()`, `Date/TimeToSec()`, `Date/SecToTime()`

1.117 DateClass/timeduration/operator*=()

DateClass/timeduration/operator*=()

NAME

`operator*=()` -- Multiply with operator for timeduration object (V33)

SYNOPSIS

```
(*this) *= tdm;

const timeduration &operator*=(const unsigned int tdm);
```

FUNCTION

Multiply the timeduration object with a constant value.

INPUTS

`*this` - timeduration object
`tdm` - Multiplication factor

RESULT

`*this` - Modified timeduration object

EXAMPLE

```
...
td *= 2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator+=()`, `operator-=()`, `operator*()`,
`operator/()`, `operator/=()`, `Date/TimeToSec()`, `Date/SecToTime()`

1.118 DateClass/timeduration/operator*=()

DateClass/timeduration/operator*=()

NAME

`operator*=()` -- Multiply with operator for timeduration object (V33)

SYNOPSIS

```
(*this) *= tdm;  
  
const timeduration &operator*=(const unsigned long tdm);
```

FUNCTION

Multiply the timeduration object with a constant value.

INPUTS

`*this` - timeduration object
`tdm` - Multiplication factor

RESULT

`*this` - Modified timeduration object

EXAMPLE

```
...  
td *= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator+=()`, `operator-=()`, `operator*()`,
`operator/()`, `operator/=()`, `Date/TimeToSec()`, `Date/SecToTime()`

1.119 DateClass/timeduration/operator+()

DateClass/timeduration/operator+()

NAME

operator+() -- Add operator for timeduration objects (V33)

SYNOPSIS

```
tdres = (*this) + td;

timeduration operator+(const timeduration &td) const;
```

FUNCTION

Add two timeduration objects to a new timeduration object.

INPUTS

```
*this - timeduration object
td    - timeduration object
```

RESULT

tdres - new timeduration object

EXAMPLE

```
...
tdres = td1 + td2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator-(), operator+(), operator-=(), timeduration(), Date/TimeToSec(),
Date/SecToTime()

1.120 DateClass/timeduration/operator+=()

DateClass/timeduration/operator+=()

NAME

`operator+=()` -- Add to operator for timeduration objects (V33)

SYNOPSIS

```
(*this) += td;

const timeduration &operator+=(const timeduration &td);
```

FUNCTION

Add timeduration object 2 to timeduration object 1.

INPUTS

`*this` - timeduration object
`td` - timeduration object

RESULT

`*this` - modified timeduration object

EXAMPLE

```
...
td1 += td2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator-=()`, `Date/TimeToSec()`, `Date/SecToTime()`

1.121 DateClass/timeduration/operator-()

DateClass/timeduration/operator-()

NAME

`operator-()` -- Sub operator for timeduration objects (V33)

SYNOPSIS

```
tdres = (*this) - td;

timeduration operator-(const timeduration &td) const;
```

FUNCTION

Difference between two timeduration objects to a new timeduration object.

INPUTS

*this - timeduration object
td - timeduration object

RESULT

tdres - new timeduration object

EXAMPLE

```
...  
tdres = td1 - td2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator+=(), operator-=(), timeduration(), Date/TimeDiff()
Date/SecToTime()

1.122 DateClass/timeduration/operator-=()

DateClass/timeduration/operator-=()

NAME

operator-=() -- Sub from operator for timeduration objects (V33)

SYNOPSIS

```
(*this) -= td;  
  
const timeduration &operator-=(const timeduration &td);
```

FUNCTION

Difference between two timeduration objects.

INPUTS

*this - timeduration object

td - timeduration object

RESULT

*this - modified timeduration object

EXAMPLE

```
...
td1 -= td2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), Date/TimeDiff(), Date/SecToTime()

1.123 DateClass/timeduration/operator/()

DateClass/timeduration/operator/()

NAME

operator/() -- Div operator for timeduration object (V33)

SYNOPSIS

```
tdres = (*this) / tdd;

timeduration operator/(const unsigned short tdd) const;
```

FUNCTION

Devide the timeeduration object by a constant value to a new timeeduration object.
If tdd is 0 you will get back an object with a duration of 0!

INPUTS

*this - timeduration object
tdd - Division factor

RESULT

tdres - new timeduration object

EXAMPLE

```
...  
tdres = td / 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator+(), operator-(), operator+=(), operator-=(), operator*(),  
operator*=(), operator/=( ), timeduration(), Date/TimeToSec(),  
Date/SecToTime()
```

1.124 DateClass/timeduration/operator/()

DateClass/timeduration/operator/()

NAME

operator/() -- Div operator for timeduration object (V33)

SYNOPSIS

```
tdres = (*this) / tdd;  
  
timeduration operator/(const unsigned int tdd) const;
```

FUNCTION

Devide the timeeduration object by a constant value to a new timeeduration object.
If tdd is 0 you will get back an object with a duration of 0!

INPUTS

```
*this - timeduration object  
tdd   - Division factor
```

RESULT

tdres - new timeduration object

EXAMPLE

```
...  
tdres = td / 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator+()`, `operator-()`, `operator+=()`, `operator-=()`, `operator*()`,
`operator*=()`, `operator/()`, `timeduration()`, `Date/TimeToSec()`,
`Date/SecToTime()`

1.125 DateClass/timeduration/operator/()

`DateClass/timeduration/operator/()`

NAME

`operator/()` -- Div operator for timeduration object (V33)

SYNOPSIS

```
tdres = (*this) / tdd;  
  
timeduration operator/(const unsigned long tdd) const;
```

FUNCTION

Devide the timeeduration object by a constant value to a new
timeeduration object.
If tdd is 0 you will get back an object with a duration of 0!

INPUTS

`*this` - timeduration object
`tdd` - Division factor

RESULT

`tdres` - new timeduration object

EXAMPLE

```
...  
tdres = td / 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator+(), operator-(), operator+==(), operator-==(), operator*(),
operator*=(), operator/==(), timeduration(), Date/TimeToSec(),
Date/SecToTime()
```

1.126 DateClass/timeduration/operator/()

DateClass/timeduration/operator/()

NAME

operator/() -- Div operator for timeduration object (V33)

SYNOPSIS

```
tdf = (*this) / td;

unsigned long operator/(const timeduration &td) const;
```

FUNCTION

Devide the timeduration object by another timeduration object to the devision factor.

INPUTS

```
*this - timeduration object
td     - timeduration object
```

RESULT

tdf - Devision factor

EXAMPLE

```
...
tdf = td1 / td2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

```
operator+(), operator-(), operator+==(), operator-==(), operator*(),
operator*=(), operator/==(), operator%(), Date/TimeToSec()
```

1.127 DateClass/timeduration/operator/=()

DateClass/timeduration/operator/=()

NAME

operator/=() -- Div through operator for timeduration object (V33)

SYNOPSIS

```
(*this) /= tdd;  
  
const timeduration &operator/=(const unsigned short tdd);
```

FUNCTION

Devide the timeduration object by a constant value.
If tdd is 0 you will get back an object with a duration of 0!

INPUTS

*this - timeduration object
tdd - Division factor

RESULT

*this - Modified timeduration object

EXAMPLE

```
...  
td /= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator*(),
operator/(), operator*=(), Date/TimeToSec(), Date/SecToTime()

1.128 DateClass/timeduration/operator/=()

DateClass/timeduration/operator/=()

NAME

operator/=() -- Div through operator for timeduration object (V33)

SYNOPSIS

```
(*this) /= tdd;

const timeduration &operator/=(const unsigned int tdd);
```

FUNCTION

Devide the timeduration object by a constant value.
If tdd is 0 you will get back an object with a duration of 0!

INPUTS

*this - timeduration object
tdd - Division factor

RESULT

*this - Modified timeduration object

EXAMPLE

```
...
td /= 2;
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator*(),
operator/(), operator*=(), Date/TimeToSec(), Date/SecToTime()

1.129 DateClass/timeduration/operator/=(())

DateClass/timeduration/operator/=(())

NAME

operator/=(()) -- Div through operator for timeduration object (V33)

SYNOPSIS

```
(*this) /= tdd;

const timeduration &operator/=(const unsigned long tdd);
```

FUNCTION

Devide the timeduration object by a constant value.
If tdd is 0 you will get back an object with a duration of 0!

INPUTS

*this - timeduration object
tdd - Division factor

RESULT

*this - Modified timeduration object

EXAMPLE

```
...  
td /= 2;  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator+(), operator-(), operator+=(), operator-=(), operator*(),
operator/(), operator*=(), Date/TimeToSec(), Date/SecToTime()

1.130 DateClass/timeduration/operator<()

DateClass/timeduration/operator<()

NAME

operator<() -- Lower than operator for timeduration objects (V33)

SYNOPSIS

```
compare = ((*this) < td);  
  
bool operator<(const timeduration &td) const;
```

FUNCTION

Compares if timeduration object 1 is lower than timeduration object 2.

INPUTS

*this - timeduration object
td - timeduration object

RESULT

compare - Boolean result of the lower than compare operation

EXAMPLE

```
...
if (td1 < td2)
    printf("<\n");
else
    printf(">=\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

operator==(), operator!=(), operator>(), operator<=(), operator>=(),
Date/Compare2Times ()

1.131 DateClass/timeduration/operator<=()

DateClass/timeduration/operator<=()

NAME

operator<=() -- Lower or equal operator for timeduration objects (V33)

SYNOPSIS

```
compare = ((*this) <= td);

bool operator<=(const timeduration &td) const;
```

FUNCTION

Compares if timeduration object 1 is lower or equal than timeduration object 2.

INPUTS

```
*this - timeduration object
td     - timeduration object
```

RESULT

compare - Boolean result of the lower or equal compare operation

EXAMPLE

```
...
if (td1 <= td2)
    printf("<=\n");
else
    printf(">\n");
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator==()`, `operator!=()`, `operator<()`, `operator>()`, `operator>=()`,
`Date/Compare2Times()`

1.132 DateClass/timeduration/operator==()

`DateClass/timeduration/operator==()`

NAME

`operator==()` -- Equal operator for timeduration objects (V33)

SYNOPSIS

```
compare = ((*this) == td);

bool operator==(const timeduration &td) const;
```

FUNCTION

Compares two timeduration objects for equality.

INPUTS

```
*this - timeduration object
td    - timeduration object
```

RESULT

`compare` - Boolean result of the equal compare operation

EXAMPLE

```
...
if (td1 == td2)
    printf("==\n");
else
    printf("!=\n");
```

...

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator!=()`, `operator<()`, `operator>()`, `operator<=()`, `operator>=()`,
`Date/Compare2Times()`

1.133 DateClass/timeduration/operator>()

`DateClass/timeduration/operator>()`

NAME

`operator>()` -- Greater than operator for timeduration objects (V33)

SYNOPSIS

```
compare = ((*this) > td);  
  
bool operator>(const timeduration &td) const;
```

FUNCTION

Compares if timeduration object 1 is greater than timeduration object 2.

INPUTS

`*this` - timeduration object
`td` - timeduration object

RESULT

`compare` - Boolean result of the greater than compare operation

EXAMPLE

```
...  
if (td1 > td2)  
    printf(">\n");  
else  
    printf("<=\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator==()`, `operator!=()`, `operator<()`, `operator<=()`, `operator>=()`,
`Date/Compare2Times()`

1.134 DateClass/timeduration/operator>=()

`DateClass/timeduration/operator>=()`

NAME

`operator>=()` -- Greater or equal operator for timeduration objs (V33)

SYNOPSIS

```
compare = ((*this) >= td);  
  
bool operator>=(const timeduration &td) const;
```

FUNCTION

Compares if timeduration object 1 is greater or equal than timeduration object 2.

INPUTS

`*this` - timeduration object
`td` - timeduration object

RESULT

`compare` - Boolean result of the greater or equal compare operation

EXAMPLE

```
...  
if (td1 >= td2)  
    printf(">=\n");  
else  
    printf("<\n");  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`operator==()`, `operator!=()`, `operator<()`, `operator>()`, `operator<=()`,
`Date/Compare2Times()`

1.135 DateClass/timeduration/timeduration()

DateClass/timeduration/timeduration()

NAME

`timeduration()` -- Constructor of timeduration class (V33)

SYNOPSIS

```
td = timeduration(jd);  
  
timeduration(float jd);
```

FUNCTION

Constructor of the timeduration class using the JD format.

INPUTS

jd - Timeduration in JD format

RESULT

td - Timeduration object

EXAMPLE

```
...  
timeduration td(0.5);  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

`timeduration()`, `Date/JDToTime()`

1.136 DateClass/timeduration/timeduration()

DateClass/timeduration/timeduration()

NAME

timeduration() -- Constructor of timeduration class (V33)

SYNOPSIS

```
td = timeduration(jd);  
  
timeduration(double jd);
```

FUNCTION

Constructor of the timeduration class using the JD format.

INPUTS

jd - Timeduration in JD format

RESULT

td - Timeduration object

EXAMPLE

```
...  
timeduration td(0.5);  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

timeduration(), Date/JDToTime()

1.137 DateClass/timeduration/timeduration()

DateClass/timeduration/timeduration()

NAME

timeduration() -- Constructor of timeduration class (V33)

SYNOPSIS

```
td = timeduration(hours=0,mins=0,secs=0);  
  
timeduration(unsigned short hours=0, unsigned short mins=0,
```

```
    unsigned short secs=0);
```

FUNCTION

Constructor of the timeduration class.

INPUTS

hours - Hours of a timeduration
mins - Mins of a timeduration
secs - Secs of a timeduration

RESULT

td - Timeduration object

EXAMPLE

```
...  
timeduration td(2,30,0);  
...
```

NOTES

None.

BUGS

No known bugs.

SEE ALSO

timeduration(), Date/SecToTime()
