C++ DATE AND TIME

The C++ standard library does not provide a proper date type. C++ inherits the structs and functions for date and time manipulation from C. To access date and time related functions and structures, you would need to include <ctime> header file in your C++ program.

There are four time-related types: **clock_t, time_t, size_t**, and **tm**. The types clock_t, size_t and time_t are capable of representing the system time and date as some sort of integer.

The structure type **tm** holds the date and time in the form of a C structure having the following elements:

```
struct tm {
  int tm_sec;   // seconds of minutes from 0 to 61
  int tm_min;   // minutes of hour from 0 to 59
  int tm_hour;  // hours of day from 0 to 24
  int tm_mday;  // day of month from 1 to 31
  int tm_mon;   // month of year from 0 to 11
  int tm_year;  // year since 1900
  int tm_wday;  // days since sunday
  int tm_yday;  // days since January 1st
  int tm_isdst; // hours of daylight savings time
}
```

Following are the important functions, which we use while working with date and time in C or C++. All these functions are part of standard C and C++ library and you can check their detail using reference to C++ standard library given below.

SN	Function & Purpose
1	**time_t time*time_t * time*;** This returns the current calendar time of the system in number of seconds elapsed since January 1, 1970. If the system has no time, .1 is returned.
2	**char *ctime*consttime_t * time*;** This returns a pointer to a string of the form *day month year hours:minutes:seconds year\n\0*.
3	**struct tm *localtime*consttime_t * time*;** This returns a pointer to the **tm** structure representing local time.
4	**clock_t clock*void*;** This returns a value that approximates the amount of time the calling program has been running. A value of .1 is returned if the time is not available.
5	**char * asctime** *conststructtm * time*;** This returns a pointer to a string that contains the information stored in the structure pointed to by time converted into the form: day month date hours:minutes:seconds year\n\0
6	**struct tm *gmtime*consttime_t * time*;**

This returns a pointer to the time in the form of a tm structure. The time is represented in Coordinated Universal Time *UTC*, which is essentially Greenwich Mean Time *GMT*.

7 **time_t mktime*structtm * time*;**

This returns the calendar-time equivalent of the time found in the structure pointed to by time.

8 **double difftime *time_ttime2, time_ttime1*;**

This function calculates the difference in seconds between time1 and time2.

9 **size_t strftime;**

This function can be used to format date and time a specific format.

Current date and time:

Consider you want to retrieve the current system date and time, either as a local time or as a Coordinated Universal Time *UTC*. Following is the example to achieve the same:

```
#include <iostream>
#include <ctime>

using namespace std;

int main( )
{
   // current date/time based on current system
   time_t now = time(0);

   // convert now to string form
   char* dt = ctime(&now);

   cout << "The local date and time is: " << dt << endl;

   // convert now to tm struct for UTC
   tm *gmtm = gmtime(&now);
   dt = asctime(gmtm);
   cout << "The UTC date and time is:"<< dt << endl;
}
```

When the above code is compiled and executed, it produces the following result:

```
The local date and time is: Sat Jan  8 20:07:41 2011

The UTC date and time is:Sun Jan  9 03:07:41 2011
```

Format time using struct tm:

The **tm** structure is very important while working with date and time in either C or C++. This structure holds the date and time in the form of a C structure as mentioned above. Most of the time related functions makes use of tm structure. Following is an example which makes use of various date and time related functions and tm structure:

While using structure in this chapter, I'm making an assumption that you have basic understanding on C structure and how to access structure members using arrow -> operator.

```
#include <iostream>
#include <ctime>
```

```
using namespace std;

int main( )
{
   // current date/time based on current system
   time_t now = time(0);

   cout << "Number of sec since January 1,1970:" << now << endl;

   tm *ltm = localtime(&now);

   // print various components of tm structure.
   cout << "Year: "<< 1900 + ltm->tm_year << endl;
   cout << "Month: "<< 1 + ltm->tm_mon<< endl;
   cout << "Day: "<<  ltm->tm_mday << endl;
   cout << "Time: "<< 1 + ltm->tm_hour << ":";
   cout << 1 + ltm->tm_min << ":";
   cout << 1 + ltm->tm_sec << endl;
}
```

When the above code is compiled and executed, it produces the following result:

```
Number of sec since January 1, 1970:1294548238
Year: 2011
Month: 1
Day: 8
Time: 22: 44:59
```