**WEEKDAYS COUNTER**

**A MINI PROJECT REPORT**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF

THE DEGREE OF

**BACHELOR OF ENGINEERING**

(COMPUTER SCIENCE AND ENGINEERING)

IN THE SUBJECT OF

**OBJECT ORIENTED PROGRAMMING WITH C++**

**SUBMITTED TO**

**M S RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE**

**SUBMITTED BY**

BINEET KUMAR 1MS13CS036

DEEPAK JAYAPRAKASH 1MS13CS037

**SUPERVISED BY**

J GEETHA REDDY

November 2014



**MS RAMAIAH INSTITUTE OF TECHNOLOGY**

**BANGALORE**

**ABSTRACT**

The concept on which the project is based upon is to count the number of times each day of the week occurs between the given two dates. This is being implemented by using a powerful feature of the standard template library and standard function library. The use of STL and SFL makes it easy for design and coding the project. The use of vector and a header file known as ctime is used in the code. These particular functions and header are chosen as they are very apt for the given implementation and makes it easier to handle all test cases.

The STL feature that is being used here is vector. This is very suitable and powerful as it can handle the size dynamically and since it was very much necessary in the execution. **TABLE OF CONTENTS**

Page No.

*Abstract* *1*

|  |  |  |
| --- | --- | --- |
| 1 | Code | 2 |
| 2 | Results | 9 |
| 3 | Conclusion | 11 |

1. **CODE**

#include<iostream>

#include<cstring>

#include<ctime>

#include<vector>

#include<conio.h> // for getch()

#include<stdlib.h> // for system("cls")

#include<stdio.h> // for fflush(stdin)

using namespace std;

char weekdays[7][4]={"Sun","Mon","Tue","Wed","Thu","Fri","Sat"}; // array to hold the names of weekdays

class date

{

int day;

int month;

int year;

public: void input(); // member functions

time\_t dateToTime();

};

void date:: input() // function to input the dates

{

while(1)

{

cout<<"\n\n\n\n\t\t Enter the date in the format :date time year:";;

cin>>day>>month>>year;

fflush(stdin);

if(year%400==0||(year%4==0&&year&100!=0)) // for leap year

{

if(month<1||month>12||(month==1&&(day<1||day>31))||(month==2&&(day<1||day>29))||(month==3&&(day<1||day>31))||(month==4&&(day<1||day>30))||(month==5&&(day<1||day>31))||(month==6&&(day<1||day>30))||(month==7&&(day<1||day>31))||(month==8&&(day<1||day>31))||(month==9&&(day<1||day>30))||(month==10&&(day<1||day>31))||(month==11&&(day<1||day>30))||(month==12&&(day<1||day>31)))

cout<<"\n\n\n\t\t entered date doesnt exist please re-enter\n\n\n";

else

return;

}

else // not a leap year

{

if(month<1||month>12||(month==1&&(day<1||day>31))||(month==2&&(day<1||day>28))||(month==3&&(day<1||day>31))||(month==4&&(day<1||day>30))||(month==5&&(day<1||day>31))||(month==6&&(day<1||day>30))||(month==7&&(day<1||day>31))||(month==8&&(day<1||day>31))||(month==9&&(day<1||day>30))||(month==10&&(day<1||day>31))||(month==11&&(day<1||day>30))||(month==12&&(day<1||day>31)))

cout<<"\t\tentered date doesnt exist please re-enter:";

else

return;

}

}

}

time\_t date:: dateToTime() //function to convert the date into time\_t format

{

tm tmp = tm(); // tm is a structure which holds the time in broken down time

tmp.tm\_mday = day;

tmp.tm\_mon = month - 1;

tmp.tm\_year = year - 1900; // we subtract 1900 because the standard format takes the year from 1900

return mktime(&tmp);

}

int get\_day(time\_t d) // function to get the day number of the week

{

int j,day\_number;

char \*c1=ctime(&d);

char day1[4];

for(j=0;j<3;j++) // copies the first 3 characters(which holds the day) of c1 into day1

day1[j]=c1[j];

day1[j]='\0'; // appemds a null character in the end

for(int k=0;k<7;k++) // the loop compares the day with the weekdays array and assigns day\_number with its respective value

{

if(strcmp(weekdays[k],day1)==0)

day\_number=k;

}

return day\_number;

}

void print\_weekends(time\_t d,int dn,int count\_sunday)

{

vector<string>sundays(count\_sunday); // allocating dynamically because count\_sunday is decided at runtime.

for(int i=0;i<1;i++)

{

if(dn>=1&&dn<=6)

d=d+86400\*(7-dn);

else if(dn==0)

d=d+86400\*7;

sundays[i]=ctime(&d);

}

for(int i=1;i<count\_sunday;i++)

{

d+=86400\*7;

sundays[i]=ctime(&d);

}

cout<<"\n\n sundays occur on the following days\n";

for(int i=0;i<count\_sunday;i++)

cout<<sundays[i];

}

void calculate(time\_t d1,time\_t d2) // calulats the number of days and displays it

{

int i;

double sec = difftime(d2, d1); // returns the difference in seconds

vector <int>count(7);

int days = (sec / (60\*60\*24)); //converts seconds to days

int weeks = days / 7; // days to weeks

int day\_number1=get\_day(d1);

int day\_number2=get\_day(d2);

if(day\_number1!=day\_number2) // handles the case when starting day and ending day are not he same

{

for(i=0;i<7;i++)

{

if(day\_number1<day\_number2)

{

if(i>day\_number1&&i<day\_number2)

count[i]=weeks+1;

else

count[i]=weeks;

}

else

{

if(i>=day\_number2&&i<=day\_number1)

count[i]=weeks;

else

count[i]=weeks+1;

}

}

}

else{ // handles the case when starting and ending day are the same

for(i=0;i<7;i++)

{

if(i==day\_number1&&weeks!=0) // we check weeks not equal to zero because number of days less than 7(i.e. weeks=0), then answer would have been -1 not 0

count[i]=weeks-1;

else

count[i]=weeks;

}

}

cout<<"\n\t\t\t\t WEEKDAY COUNT\n\n";

cout<<"===============================================================================";

cout<<"\n \n total number of days:\t"<<days;

cout<<"\n\n\n number of weeks="<<weeks;

cout<<"\n\n\nbetween the entered two dates the number of times each day occurs is as follows:\n\n\n";

cout<<"\n\n Day \t Occurence";

for(i=0;i<7;i++)

cout<<"\n"<<weekdays[i]<<"\t"<<count[i];

cout<<"\n\n \t\t the number of weekends( saturdays and sundays) is: "<<count[0]+count[6];

cout<<"\n\n \t\t the number of working days is : "<<count[1]+count[2]+count[3]+count[4]+count[5]-count[6]-count[0]<<"\n\n\n";

print\_weekends(d1,day\_number1,count[0]);

cout<<"\t\t press any key to continue";

getch();

}

int main()

{

int choice=1;

cout<<"\n\n\n\n===============================================================================";

cout<<"\n\n\n\n \t\t\t\t WEEKDAY COUNTER\n\n\n\n\n\n\t\t\tpress any key to continue.......";

cout<<"\n\n\n\n===============================================================================";

getch();

time\_t d1; // time\_t variables

time\_t d2;

date date1,date2; // objects

do

{

system("cls");

date1.input();

system("cls");

date2.input();

system("cls");

d1=date1.dateToTime();

d2=date2.dateToTime();

calculate(d1,d2);

system("cls");

cout<<"\n\n\n\n\t\t\t press\n\t\t\t1. if you want to continue\n\n\t\t\t2. if you want to exit\n\n\t\t\t enter your choice:";

cin>>choice;

while(choice!=1&&choice!=2)

{

cout<<"\n\n\t entered choice is wrong:(...please re-enter:";

cin>>choice;

}

}while(choice==1);

system("cls");

cout<<"\n\n\n\n\t\t\t\tTHANK YOU\n\n\n";

return 0;

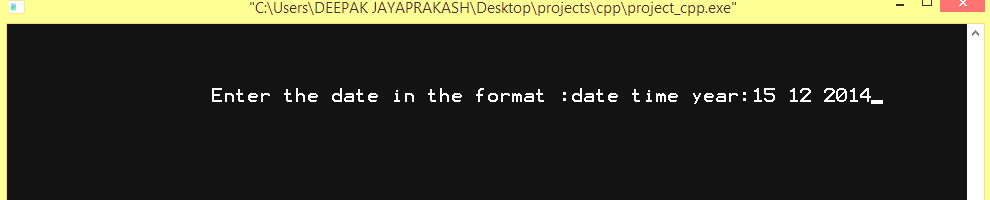
}

1. **RESULTS**

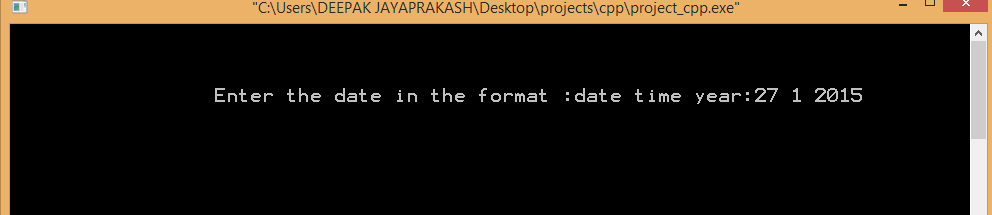
**Sample Input and Output screens:**

**INPUT:**

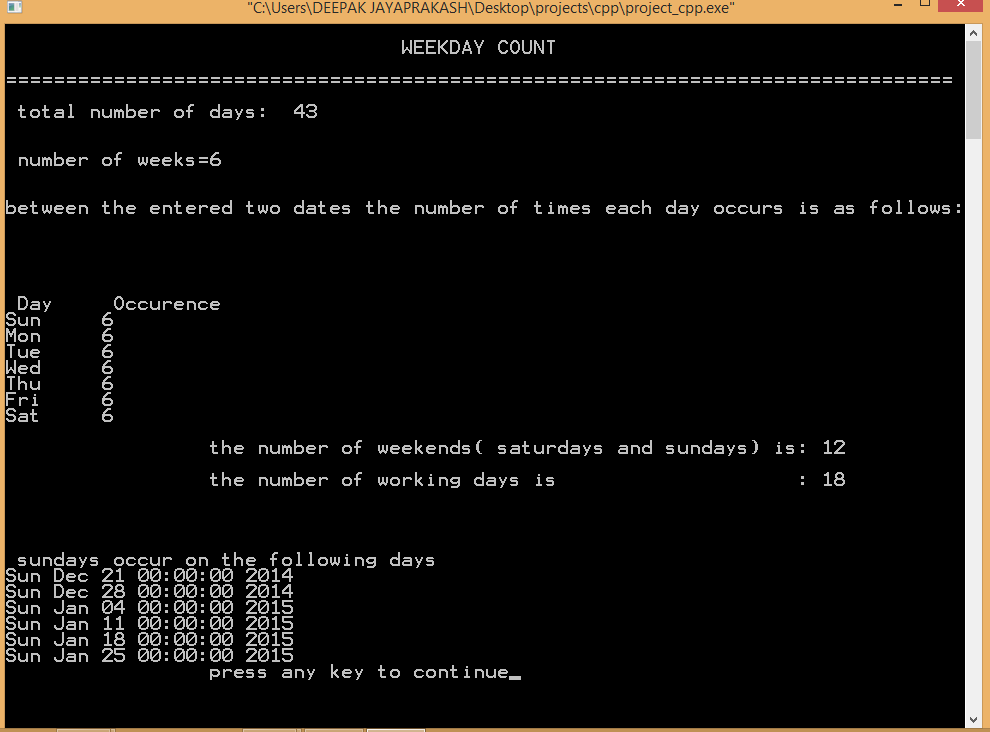
**1.**



2.



OUTPUT:



**3.CONCLUSION**

The project is implemented to illustrate the usage of STL features in our code to demonstrate the fact that it is in fact a powerful concept of C++. It gave us a clear understanding of the various functions and provisions that we have and how we can make use of it to use that in some real time applications. By making use of

in-built sophisticated STL features, we not only reduce the size of the code but also make it clear and much less complicated.

The application of STL to develop some real time problem really helped us to understand the need for such functions and how we can make the best use of them to extend this furthermore.