**/\* Program No. :**

**Aim : WAP to create a class Time consisting of member variables hour, minute, second and implement the concept of operator overloading for the following operations : Addition, Subtraction, Post Increment, Post Decrement, Pre Increment, Pre Decrement, Comparison (Less Than and Greater Than), Assignment Addition, Assignment Subtraction.**

**\*/**

#include<iostream.h>

#include<conio.h>

enum bool{false=0,true};

class time

{

int hr,min,sec;

public:

time();

time(int,int,int);

time operator + (time);

time operator - (time);

time operator ++ (int);

time operator -- (int);

time operator ++ ();

time operator -- ();

bool operator > (time);

bool operator < (time);

time operator += (time);

time operator -= (time);

void display();

};

time::time()

{

hr=0;

min=0;

sec=0;

}

time::time(int x,int y,int z)

{

hr=x;

min=y;

sec=z;

}

time time::operator + (time x)

{

time ans;

ans.hr=hr+x.hr;

ans.min=min+x.min;

ans.sec=sec+x.sec;

if(ans.sec>=60)

{

ans.sec-=60;

ans.min++;

}

if(ans.min>=60)

{

ans.min-=60;

ans.hr++;

}

return(ans);

}

time time::operator - (time x)

{

time ans;

ans.hr=hr-x.hr;

ans.min=min-x.min;

ans.sec=sec-x.sec;

if(ans.sec<0)

{

ans.sec+=60;

ans.min--;

}

if(ans.min<0)

{

ans.min+=60;

ans.hr--;

}

return(ans);

}

time time::operator ++ (int)

{

time ans;

ans.hr=hr;

ans.min=min;

ans.sec=sec++;

if(sec>=60)

{

sec-=60;

min++;

}

if(min>=60)

{

min-=60;

hr++;

}

return(ans);

}

time time::operator -- (int)

{

time ans;

ans.hr=hr;

ans.min=min;

ans.sec=sec--;

if(sec<0)

{

sec+=60;

min--;

}

if(min<0)

{

min+=60;

hr--;

}

return(ans);

}

time time::operator ++ ()

{

++sec;

if(sec>=60)

{

sec-=60;

min++;

}

if(min>=60)

{

min-=60;

hr++;

}

return(time(hr,min,sec));

}

time time::operator -- ()

{

--sec;

if(sec<0)

{

sec+=60;

min--;

}

if(min<0)

{

min+=60;

hr--;

}

return(time(hr,min,sec));

}

bool time::operator > (time x)

{

if(hr>x.hr)

return(true);

if(hr==x.hr&&min>x.min)

return(true);

if(hr==x.hr&&min==x.min&&sec>x.sec)

return(true);

else

return(false);

}

bool time::operator < (time x)

{

if(hr<x.hr)

return(true);

if(hr==x.hr&&min<x.min)

return(true);

if(hr==x.hr&&min==x.min&&sec<x.sec)

return(true);

else

return(false);

}

time time::operator += (time x)

{

hr+=x.hr;

min+=x.min;

sec+=x.sec;

if(sec>=60)

{

sec-=60;

min++;

}

if(min>=60)

{

min-=60;

hr++;

}

return(time(hr,min,sec));

}

time time::operator -= (time x)

{

time ans;

hr-=x.hr;

min-=x.min;

sec-=x.sec;

if(sec<0)

{

sec+=60;

min--;

}

if(min<0)

{

min+=60;

hr--;

}

return(time(hr,min,sec));

}

void time::display()

{

cout<<hr<<" hr "<<min<<" min "<<sec<<" sec";

}

void main()

{

int a,b,c,x,y,z,choice;

char choice2;

bool ans;

do

{

clrscr();

cout<<"\n\n\t\t\t\tMENU\n\n\t\t1. Addition\n\t\t2. Subtraction"

<<"\n\t\t3. Post Increment\n\t\t4. Post Decrement"

<<"\n\t\t5. Pre Increment\n\t\t6. Pre Decrement"

<<"\n\t\t7. Comparison\n\t\t8. Assignment Addition"

<<"\n\t\t9. Assignment Subtraction";

cout<<"\n\n\tEnter your choice (1-9) : ";

cin>>choice;

if(choice==3||choice==4||choice==5||choice==6)

{

cout<<"\n\n\n\tEnter Time 1 (hr) : ";

cin>>a;

cout<<"\t\t (min) : ";

cin>>b;

cout<<"\t\t (sec) : ";

cin>>c;

}

else

{

cout<<"\n\n\n\tEnter Time 1 (hr) : ";

cin>>a;

cout<<"\t\t (min) : ";

cin>>b;

cout<<"\t\t (sec) : ";

cin>>c;

cout<<"\n\tEnter Time 2 (hr) : ";

cin>>x;

cout<<"\t\t (min) : ";

cin>>y;

cout<<"\t\t (sec) : ";

cin>>z;

}

time t1(a,b,c),t2(x,y,z),t3;

switch(choice)

{

case 1:t3=t1+t2;

cout<<"\n\n\tTime 3 = Time 1 + Time 2"

<<"\n\n\tTime 3 = ";

t3.display();

break;

case 2:t3=t1-t2;

cout<<"\n\n\tTime 3 = Time 1 - Time 2"

<<"\n\n\tTime 3 = ";

t3.display();

break;

case 3:t3=t1++;

cout<<"\n\n\tTime 3 = Time 1 ++"

<<"\n\n\tTime 3 = ";

t3.display();

break;

case 4:t3=t1--;

cout<<"\n\n\tTime 3 = Time 1 --"

<<"\n\n\ttime 3 = ";

t3.display();

break;

case 5:t3=++t1;

cout<<"\n\n\tTime 3 = ++ Time 1"

<<"\n\n\tTime 3 = ";

t3.display();

break;

case 6:t3=--t1;

cout<<"\n\n\tTime 3 = -- Time 1"

<<"\n\n\tTime 3 = ";

t3.display();

break;

case 7:ans=t1<t2;

cout<<"\n\n\tAns = Time 1 < Time 2"

<<"\n\n\tAns = ";

if(ans==true)

cout<<"True";

else

cout<<"False";

ans=t1>t2;

cout<<"\n\n\tAns = Time 1 > Time 2"

<<"\n\n\tAns = ";

if(ans==true)

cout<<"True";

else

cout<<"False";

break;

case 8:t3=t1+=t2;

cout<<"\n\n\tTime 3 = Time 1 += Time 2"

<<"\n\n\tTime 3 = ";

t3.display();

break;

case 9:t3=t1-=t2;

cout<<"\n\n\tTime 3 = Time 1 -= Time 2"

<<"\n\n\tTime 3 = ";

t3.display();

break;

default:cout<<"\n\n\tInvalid Choice";

}

cout<<"\n\n\tWant to continue (y/n) : ";

cin>>choice2;

}while(choice2=='Y'||choice2=='y');

getch();

}

**/\***

**Name : Rohit Aggarwal**

**Roll No. : 7CS-097**

**\*/**