Course Name: A Level (2nd Sem) **Subject:** Data Struture using C++
Topic: Example of Operator Overloading **Date:** 03-04-2020

Example of Operator Overloading

Example 1:

```
#include <iostream.h>
#include <conio.h>
class Time
{
   int h,m,s;
   public:
   Time()
   {
      h=0, m=0; s=0;
   }
   void setTime();
   void show()
   {
      cout<< h<< ":"<< m<< ":"<< s;
   }
    //overloading '+' operator
   Time operator+(time);
};

Time Time::operator+(Time t1)        //operator function
{
   Time t;
   int a,b;
   a = s+t1.s;
   t.s = a%60;
   b = (a/60)+m+t1.m;
   t.m = b%60;
   t.h = (b/60)+h+t1.h;
   t.h = t.h%12;
   return t;
}

void time::setTime()
{
   cout << "\n Enter the hour(0-11) ";
   cin >> h;
   cout << "\n Enter the minute(0-59) ";
   cin >> m;
   cout << "\n Enter the second(0-59) ";
   cin >> s;
}

void main()
{
   Time t1,t2,t3;
    cout << "\n Enter the first time ";
```

```
    t1.setTime();
    cout << "\n Enter the second time ";
    t2.setTime();
    t3 = t1 + t2; //adding of two time object using '+' operator
    cout << "\n First time ";
    t1.show();
    cout << "\n Second time ";
    t2.show();
    cout << "\n Sum of times ";
    t3.show();
    getch();
}
```

Output

```
1:20:30
2:15:25
3:35:55
```

Example 2:

```
#include <iostream>
using namespace std;
class MinusOverload {
private:
 int a;
 int b;
public:
 void Distance()
 {
 a = 0;
 b = 0;
 }
 MinusOverload(int f, int i)
 {
 int c;
 a = f;
 b = i;
 c = a - b;
 cout << "\nC:" << c;
 }
 void display()
 {
 cout << "A: " << a << " B:" << b << endl;
 }
 MinusOverload operator-()
 {
 a = -a;
 b = -b;
 return MinusOverload(a, b);
 }
};
int main()
{
 MinusOverload M1(6, 8), M2(-3, -4);
 -M1;
 M1.display();
 -M2;
```

```
 M2.display();
 return 0;
}
```

Example 3:

```
class Time
{
   int hr, min, sec;
   public:
   // default constructor
   Time()
   {
      hr=0, min=0; sec=0;
   }

   // overloaded constructor
   Time(int h, int m, int s)
   {
      hr=h, min=m; sec=s;
   }

   //overloading '==' operator
   friend bool operator==(Time &t1, Time &t2);
};

/*
   Defining the overloading operator function
   Here we are simply comparing the hour, minute and
   second values of two different Time objects to compare
   their values
*/
bool operator== (Time &t1, Time &t2)
{
   return ( t1.hr == t2.hr && t1.min == t2.min && t1.sec == t2.sec );
}

void main()
{
   Time t1(3,15,45);
   Time t2(4,15,45);
   if(t1 == t2)
   {
      cout << "Both the time values are equal";
   }
   else
   {
      cout << "Both the time values are not equal";
   }
}
```

Output:

Both the time values are not equal