Institute of Engineering & Management
Department of Computer Science & Engineering
Programming Practices Using C++ Lab for 3rd year 5th semester 2018
Code: CS593

Date: 30/08/18

WEEK-4

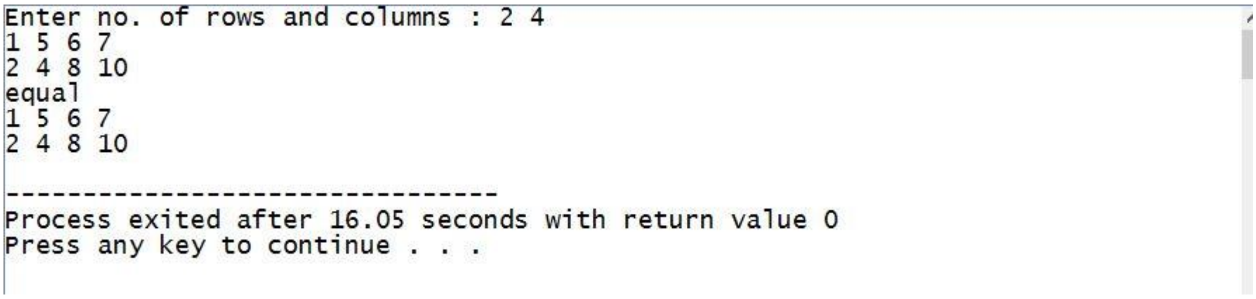
Assignment-1

Problem Statement: Write a program to declare two 2d vectors v1 and v2. Set the content of v1 = v2 by overloading the = operator. Finally check whether the content of two 2d vectors are equal or not by overloading the == operator.

Source code:

```
#include<iostream>
#include<string>
#include<vector>
template<typename T>
struct C
{
  std::vector<std::vector <T>>vect;
  C()
  {
        T value;
        for (inti = 0; i< 3; i++)
        {
              std::vector <T> temp;
              for (int j = 0; j < 3;j++)
              {
                    std::cin>>value;
                    temp.push_back(value);
              }
              vect.push_back(temp);
        }
  }
  C(T rows,T cols)
  {
        T value;
        for (inti = 0; i< rows; i++)
        {
              std::vector <T> temp;
              for (int j = 0; j <cols;j++)
              {
                    std::cin>>value;
                    temp.push_back(value);
              }
              vect.push_back(temp);
        }
  }
  void operator= (C<T>& v)
  {
        for(inti=0;i<v.vect.size();i++)
        {
              for(int j=0;j<v.vect[i].size();j++)
                    vect[i][j]=v.vect[i][j];
        }
  }
  bool operator== (C<T>& v)
  {
        if (vect==v.vect)
              return true;
  }
};
int main()
{
  introws,cols;
  std::cout<<"Enter no. of rows and columns : ";
  std::cin>>rows>>cols;
  C<int>c1(rows,cols);
  C<int> c2 = c1;
  if(c1==c2)
        std::cout<<"equal\n";
  for(auto& i:c2.vect){
        for (auto&j :i)
              {std::cout<<j<<" ";}
  std::cout<<std::endl;
}
```

Screen-Shot:



```
Enter no. of rows and columns : 2 4
1 5 6 7
2 4 8 10
equal
1 5 6 7
2 4 8 10

--------------------------------
Process exited after 16.05 seconds with return value 0
Press any key to continue . . .
```

Assignment-2

Problem Statement: Create a class/struct named time which represents the time. This should have three variables for setting the time in hours, minutes and seconds. Constructors should be used to initialize these values.

a. Add a method display() which should display the current time.
b. Overload the '+' operator to add two time objects based on a 24 hour clock. Overload the ' < ' operator to compare two time objects.

Source code:

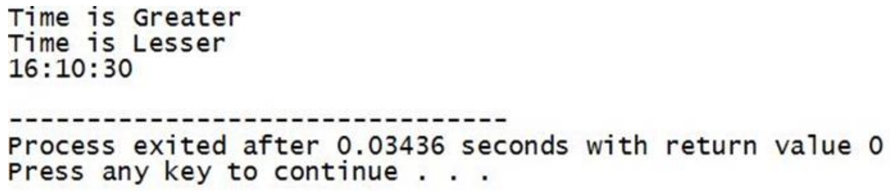
```
#include<iostream>
#include<string>
#include<vector>
template<typename T>
struct Time
{
  int hours;
  int minutes;
  int seconds;

  Time()
  {
        hours=12;
        minutes=15;
        seconds=0;
  }
```

```
    Time(inth,intm,int s)
    {
        hours=h;
        minutes=m;
        seconds=s;
    }
    Time <T> operator+ (Time<T>& v)
    {
        Time<T> c1;
        c1.hours=(hours+v.hours)%24;
        c1.minutes=minutes+v.minutes;
        if (c1.minutes/60 != 0)
        {
            c1.hours += c1.minutes/60;
            c1.minutes = c1.minutes%60;
        }
        c1.seconds=seconds+v.seconds;
        if (c1.seconds/60 != 0)
        {
            c1.minutes += c1.seconds/60;
            c1.seconds = c1.seconds%60;
        }
        return c1;
    }

    void operator> (Time<T>& v)
    {
        if ((hours*60*60+minutes*60+seconds) >
                                 (v.hours*60*60+v.minutes*60+seconds))
        {
            std::cout<<"Time is Greater"<<std::endl;
        }
        else
        {
            std::cout<<"Time is Lesser "<<std::endl;
        }
    }
    void show()
    {
        std::cout<<hours<<":"<<minutes<<":"<<seconds<<std::endl;
    }
};
int main()
{
    Time<int>c1(3,55,30);
    Time<int> c2;
    c2>c1;
    c1>c2;
    Time<int> c3=c1+c2;
    c3.show();
}
```

Screen-Shot:



```
Time is Greater
Time is Lesser
16:10:30

--------------------------------
Process exited after 0.03436 seconds with return value 0
Press any key to continue . . .
```