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

Date: 03/03/18

WEEK-2

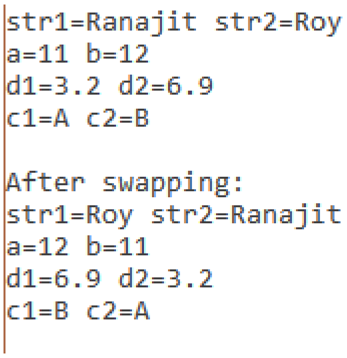
Assignment-1

Problem Statement: Write a program in C++ to create a generic swap function using templates.

Source code:

```
#include <iostream>
#include <string>

template <typename T>

void swap(T& a, T& b)
{
  T temp = a;
  a = b;
  b = temp;
}

int main()
{
  std::string str1 = "Ranajit", str2 = "Roy";
  int a=11, b=12;
  double d1 = 3.2, d2= 6.9;
  char c1='A', c2='B';
  std::cout<<"str1="<<str1<<" str2="<<str2<<std::endl;
  std::cout<<"a="<<a<<" b="<<b<<std::endl;
  std::cout<<"d1="<<d1<<" d2="<<d2<<std::endl;
  std::cout<<"c1="<<c1<<" c2="<<c2<<std::endl;
  std::cout<<"\nAfter swapping:\n";
  swap(str1, str2);
  std::cout<<"str1="<<str1<<" str2="<<str2<<std::endl;
  swap(a, b);
  std::cout<<"a="<<a<<" b="<<b<<std::endl;
  swap(d1, d2);
  std::cout<<"d1="<<d1<<" d2="<<d2<<std::endl;
  swap(c1, c2);
  std::cout<<"c1="<<c1<<" c2="<<c2<<std::endl;
  return 0;
}
```

Screen-Shot:



```
str1=Ranajit str2=Roy
a=11 b=12
d1=3.2 d2=6.9
c1=A c2=B

After swapping:
str1=Roy str2=Ranajit
a=12 b=11
d1=6.9 d2=3.2
c1=B c2=A
```

Assignment-2

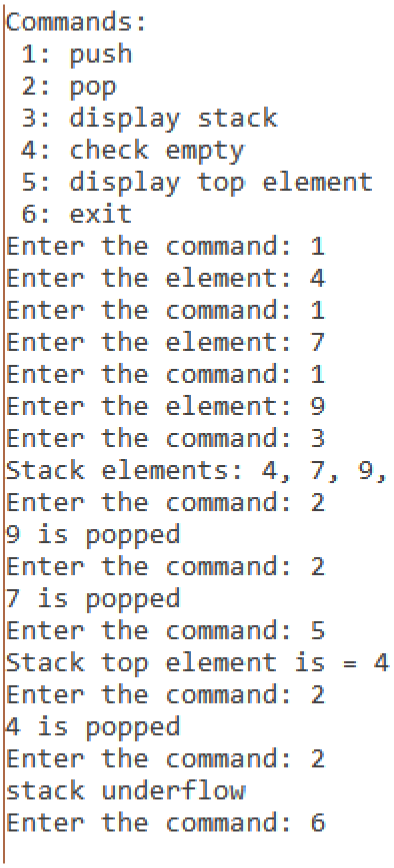
Problem Statement: Implement a stack using vectors. It should contain functions for inserting an element, deleting an element, displaying the content, check whether the stack is empty or not, displaying the top element of the stack.

Source code:

```
#include <iostream>
#include <vector>

#define MAX 100

class Stack
{
  std::vector<int> stk;
  public:
  void push(int elm)
  {
        if(MAX == stk.size())
              std::cout<<"stack overflow\n";
        else stk.push_back(elm);
  }
  void pop()
  {
        if(stk.size()==0)
              std::cout<<"\rstack underflow\n";
        else{
              std::cout<<"\r"<<stk[stk.size()-1]<<" is popped\n";
              stk.pop_back();
        }
  }
  void display()
  {
        if(stk.size()==0)
              std::cout<<"\rempty stack\n";
        else{
              std::cout<<"\rStack elements:";
              for(int i=0;i<stk.size();i++)
                    std::cout<<" "<<stk[i]<<",";
              std::cout<<"\n";
        }
  }
  void display_top()
  {
        if(stk.size()==0)
              std::cout<<"\rempty stack\n";
        else  std::cout<<"\rStack top element is="<<stk[stk.size()-
                                                  1]<<"\n";
  }
  void empty()
  {
        if(stk.size()==0)
              std::cout<<"\rEmpty!\n";
        else std::cout<<"\rNot empty! Stack size is="<<stk.size()<<"\n";
  }
};

int main()
{
  int com, flag=0, elm;
```

```
    Stack st;
    std::cout<<"Commands:\n"<<" 1: push\n"<<" 2: pop\n"<<" 3: display
          stack\n"<<" 4: check empty\n"<<" 5: display top element\n"<<" 6:
                    exit\n";
    do
    {
          std::cout<<"Enter the command: ";
          std::cin>>com;
          switch(com)
          {
                case 1: std::cout<<"\rEnter the element: ";
                        std::cin>>elm;
                        st.push(elm);
                        break;
                case 2: st.pop(); break;
                case 3: st.display(); break;
                case 4: st.empty(); break;
                case 5: st.display_top(); break;
                case 6: flag=1; break;
                default: std::cout<<"\rinvalid command!!\n";
          }
    }while(flag!=1);
    return 0;
}
```

Screen-Shot:



```
Commands:
 1: push
 2: pop
 3: display stack
 4: check empty
 5: display top element
 6: exit
Enter the command: 1
Enter the element: 4
Enter the command: 1
Enter the element: 7
Enter the command: 1
Enter the element: 9
Enter the command: 3
Stack elements: 4, 7, 9,
Enter the command: 2
9 is popped
Enter the command: 2
7 is popped
Enter the command: 5
Stack top element is = 4
Enter the command: 2
4 is popped
Enter the command: 2
stack underflow
Enter the command: 6
```

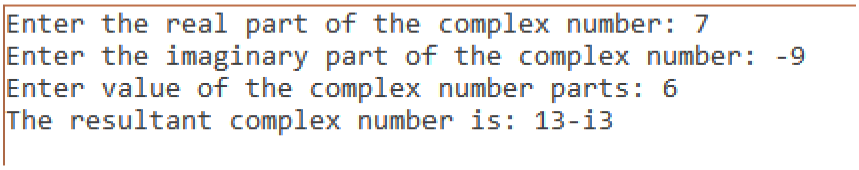
Assignment-3

Problem Statement: Write a C++ program to add two complex numbers using a class containing three constructor, a)with no parameter, b)with 1 parameter(real and imaginary parts are same), c)with 2 parameter. And two functions, a) to add the two complex numbers. b) to display the results.

Source code:

```
#include<iostream>

class Complex
{
  public:
        int real,img;
        Complex(){
              real=0;
              img=0;
        }
        Complex(int n1){
              real=n1;
              img=n1;
        }
        Complex(int n1, int n2){
              real=n1;
              img=n2;
        }
        Complex add(Complex &a,Complex &b){
              Complex temp;
              temp.real=b.real+a.real;
              temp.img=b.img+a.img;
              return temp;
        }
};
int main()
{
  int n, m;
  std::cout<<"Enter the real part of the complex number: ";
  std::cin>>n;
  std::cout<<"Enter the imaginary part of the complex number: ";
  std::cin>>m;
  Complex ob1(n,m);
  std::cout<<"Enter value of the complex number parts: ";
  std::cin>>n;
  Complex ob2(n);
  Complex ob3;
  ob3=ob3.add(ob1,ob2);
  std::cout<<"The resultant complex number is:
              "<<ob3.real<<((ob3.img<0)?"-i":"+i")<<-ob3.img<<"\n";
  return 0;
}
```

Screen-Shot:



```
Enter the real part of the complex number: 7
Enter the imaginary part of the complex number: -9
Enter value of the complex number parts: 6
The resultant complex number is: 13-i3
```