Institute of Engineering & Management
Department of Computer Science & Engineering
Object Oriented Programming (IT) Lab for 3rd year 5th semester 2018
Code: CS594D

Date: 17/07/18

WEEK-2

Assignment-1

Problem Statement: Write a Java program for addition and multiplication of complex numbers

Source Code:

```
import java.util.Scanner;

class cCal
{
public static void main(String args[])
{
      int n1,n2;
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the 1st complex number");
      n1 = sc.nextInt(); n2 = sc.nextInt();
      complex c1 = new complex(n1, n2);
      System.out.println("Enter the 2nd complex number");
      n1 = sc.nextInt(); n2 = sc.nextInt();
      complex c2 = new complex(n1, n2);
      System.out.print("Section which operation:\n 1:
                  Addition\n 2: Multiplication\nEnter: ");
      n1 = sc.nextInt();
      sc.close();
      if(n1 == 1)
            c1.add(c2);
      else c1.mult(c2);
      if(c1.imag >= 0)
            System.out.println("The resultant complex
                        number: "+c1.real+" + "+c1.imag+"i");
      else System.out.println("The resultant complex number:
                        "+c1.real+" - "+(-c1.imag)+"i");
}
}

class complex
{
int real, imag;
complex(int n1, int n2)
{
      real = n1;
      imag = n2;
}
void add(complex c)
{
      real += c.real;
      imag += c.imag;
}
```

```
        void mult(complex c)
        {
              int temp = real;
              real = real*c.real - imag*c.imag;
              imag = temp*c.imag + imag*c.real;
        }
        }
```

Screen-Shots:

```
rana@rana:~/Desktop/Java/2/complex$ java cCal
Enter the 1st complex number
2 -3
Enter the 2nd complex number
-7 -5
Section which operation:
 1: Addition
 2: Multiplication
Enter: 1
The resultant complex number: -5 - 8i
rana@rana:~/Desktop/Java/2/complex$
```

Fig: Addition

```
rana@rana:~/Desktop/Java/2/complex$ javac cCal.java
rana@rana:~/Desktop/Java/2/complex$ java cCal
Enter the 1st complex number
2 3
Enter the 2nd complex number
4 5
Section which operation:
 1: Addition
 2: Multiplication
Enter: 2
The resultant complex number: -7 + 22i
rana@rana:~/Desktop/Java/2/complex$
```

Fig: Multiplication

Assignment-2

Problem Statement: Write a Java program to implement a stack.

Source code:

```
import java.util.Scanner;

class StkImp
{
 public static void main(String[] args)
 {
       int n=0;
       Scanner sc = new Scanner(System.in);
       Stack st = new Stack();
       System.out.println("         ---- Stack ----\n");
       System.out.print("Enter the following commands:\n 1:push\n
                        2:pop\n 3:display\n 4:exit\n");
       do {
             System.out.print("Enter command: ");
             n = sc.nextInt();
             switch(n)
             {
                   case 1: System.out.print("Enter the element: ");
                           st.push(sc.nextInt()); break;
                   case 2: st.pop(); break;
                   case 3: st.display(); break;
                   case 4: System.out.println("Bye!"); break;
                   default:System.out.print("Again ");
             }
       }while(n!=4);
       sc.close();
 }
}

class Stack
{
 int top = -1;
 int[] arr = new int[100];
 void push(int n)
 {
       if(top<100)
             arr[++top] = n;
       else System.out.println("Stack overflow");
 }
 void pop()
 {
       if(top>-1)
             top--;
       else System.out.println("Stack underflow");
 }
 void display()
 {
       if(top == -1)
             System.out.println("No elemnets to print");
       else
       {
             System.out.print("The elements are: ");
```

```
                for(int i=0; i<=top; i++)
                {
                        System.out.print(arr[i]+", ");
                }
                System.out.println();
        }
    }
}
```

Screen-Shot:

```
rana@rana:~/Desktop/Java/2/stack$ javac StkImp.java
rana@rana:~/Desktop/Java/2/stack$ java StkImp
         ---- Stack ----

Enter the following commands:
 1:push
 2:pop
 3:display
 4:exit
Enter command: 1
Enter the element: 4
Enter command: 1
Enter the element: 6
Enter command: 3
The elements are: 4, 6,
Enter command: 2
Enter command: 3
The elements are: 4,
Enter command: 4
Bye!
rana@rana:~/Desktop/Java/2/stack$
```

Fig: sample I/O (1)

```
rana@rana:~/Desktop/Java/2/stack$ java StkImp
         ---- Stack ----

Enter the following commands:
 1:push
 2:pop
 3:display
 4:exit
Enter command: 1
Enter the element: 99
Enter command: 3
The elements are: 99,
Enter command: 2
Enter command: 2
Stack underflow
Enter command: 3
No elemnets to print
Enter command: 4
Bye!
rana@rana:~/Desktop/Java/2/stack$
```

Fig: sample I/O (2)