Panipat Institute of Engineering & Technology,
Samalkha (Haryana)
Computer Science & Engineering Department

Essential of Information Technology Lab

PC-CS-311LA

Submitted to:
Gaurav Gambhir Sir
(CSE Department)

Submitted by:
Anmol Baranwal
2820208
B.Tech CSE 5th Sem A3

Affiliated to

Kurukshetra University Kurukshetra, India

S no.	Name of the Program	Date	Signature
1	Write a Java Package with Stack and queue classes.		
2	Design a class for Complex numbers in Java .In addition to methods for basic operations on complex numbers, provide a method to return the number of active objects created.		
3	Develop with suitable hierarchy, class for point, shape rectangle, square, circle, ellipse, triangle, polygenetic.		
4	Design a simple test application to demonstrate dynamic polymorphism.		
5	Design a java interface for ADT Stack.		
6	Develop two different classes that implement this interface. One using array and other using linked list.		
7.	Develop a simple paint like program that can draw basic graphical primitives.		
8.	Develop a scientific calculator using event driven programming.		
9.	Develop a template for linked list class along with its members in Java.		
10.	Write a program to insert and view data using Servlets.		

Practical 1.
Write a Java Package with Stack and queue classes.

Program (Stack):-

```
import java.util.*;

public class StackDemo
{
   static void stackPush(Stack<Integer> stack){  // Stack<Integer> ensures that it receives the object type of integer rather than a usual integer
      System.out.println("\nPush Operation: \n ");
      for(int i=0;i<5;i++){
         stack.push(i);
         System.out.println("Element "+ i +" pushed into stack");
      }
      System.out.print("\n");

   }

   static void stackPop(Stack<Integer> stack){
      System.out.print("Popped Out: ");
      Integer y = (Integer) stack.pop();
      System.out.println(y);
      System.out.print("\n");
   }
   static void stackSearch(Stack<Integer> stack, int key){
      Integer pos = (Integer) stack.search(key);

      if(pos==-1) System.out.println("Element is found at position: " + pos);
      else    System.out.println("Element is found at position: " + pos);
   }
   static void stackPeek(Stack<Integer> stack)
   {
      Integer key = (Integer) stack.peek();
      System.out.println("Element on stack top: " + key);
   }

   public static void main(String[] args){
      System.out.println("\n\nAnmol Baranwal -- 2820208\n");

      Stack<Integer> stackObj = new Stack<Integer>();
      stackPush(stackObj);
      stackPop(stackObj);
      stackSearch(stackObj, 2);
      stackPeek(stackObj);
   }
}
```

OUTPUT:

```
Anmol Baranwal -- 2820208


Push Operation:

Element 0 pushed into stack
Element 1 pushed into stack
Element 2 pushed into stack
Element 3 pushed into stack
Element 4 pushed into stack

Popped Out: 4

Element is found at position: 2
Element on stack top: 3
PS E:\coding\JavaLab\Stack&QueueModule>
```

Program (Queue):-

```
import java.util.*;

public class queueMods
{
   public static void main(String args[]){

   System.out.println("\n\nAnmol Baranwal – 2820208\n\n");

   PriorityQueue<String> pq =new PriorityQueue<String>();
   pq.add("Agra");
   pq.add("Varanasi");
   pq.add("Kanpur");
   pq.add("Allahabad");
   pq.add("Lucknow");

   System.out.println("head: "+pq.element());
   System.out.println("head: "+pq.peek());
   System.out.println("iterating the queue elements:");
   Iterator itr=pq.iterator();
   while(itr.hasNext()){
     System.out.println(itr.next());
   }
   System.out.println("element removed from remove method: "+pq.remove());
   System.out.println("element removed from remove method: "+pq.poll());

   System.out.println("after removing two elements:");

   Iterator<String> itr2=pq.iterator();
   while(itr2.hasNext()){
     System.out.println(itr2.next());
   }
  }
}
```

OUTPUT:

```
Anmol Baranwal -- 2820208


head: Agra
head: Agra
iterating the queue elements:
Agra
Allahabad
Kanpur
Varanasi
Lucknow
element removed from remove method: Agra
element removed from remove method: Allahabad
after removing two elements:
Kanpur
Lucknow
Varanasi
PS E:\coding\JavaLab\Stack&QueueModule>
```

Practical 2.

Design a class for Complex numbers in Java .In addition to methods for basic operations on complex numbers, provide a method to return the number of active objects created.

Program:-

```
import java.io.*;

class ComplexNumber {
   int num1, num2;
   public static int cnt=0;
   public ComplexNumber(int x, int y) {
      num1 = x;
      num2 = y;
      cnt++;
   }
   public static String add(ComplexNumber a, ComplexNumber b) {
      int a1 = a.num1 + b.num1;
      int b1 = a.num2 + b.num2;
      if (b1 < 0)
         return (a1 + "" + b1 + "i");
      else
         return (a1 + "+" + b1 + "i");
   }
   public static String sub(ComplexNumber a, ComplexNumber b) {
      int a1 = a.num1 - b.num1;
      int b1 = a.num2 - b.num2;
      if (b1 < 0)
         return (a1 + "" + b1 + "i");
      else
         return (a1 + "+" + b1 + "i");
   }
   public static String mul(ComplexNumber a, ComplexNumber b) {

      int a1 = a.num1 * b.num1;
      int b1 = a.num2 * b.num2;
      int v1 = a.num1 * b.num2;
      int v2 = b.num1 * a.num2;
      int vi = v1 + v2;
      if (vi < 0)
         return (a1 - b1 + "" + vi + "i");
      else
         return (a1 - b1 + "+" + vi + "i");
   }
}

class ComplexNo{
   public static void main(String[] args) throws IOException {
```

```
        System.out.println("\n\nAnmol Baranwal -- 2820208\n");

        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter two numbers for complex number 1: ");
        int x = Integer.parseInt( in.readLine());
        int y = Integer.parseInt( in.readLine());

        System.out.println("Enter two numbers for complex number 2: ");
        int m = Integer.parseInt( in.readLine());
        int n = Integer.parseInt( in.readLine());

        ComplexNumber c1 = new ComplexNumber(x, y);
        ComplexNumber c2 = new ComplexNumber(m, n);

        System.out.println("\nAddition of complex numbers: " + ComplexNumber.add(c1, c2));
        System.out.println("\nSubtraction of complex numbers: " + ComplexNumber.sub(c1, c2));
        System.out.println("\nMultiplication of complex numbers: " + ComplexNumber.mul(c1, c2));
        System.out.println("\nTotal no of objects created: " + ComplexNumber.cnt +"\n");
    }
}
```

OUTPUT:

```
Anmol Baranwal -- 2820208

Enter two numbers for complex number 1:
3
-5
Enter two numbers for complex number 2:
4
-2

Addition of complex numbers: 7-7i

Subtraction of complex numbers: -1-3i

Multiplication of complex numbers: 2-26i

Total no of objects created: 2

PS E:\coding\JavaLab\Complex Number>
```

Practical 3.

Develop with suitable hierarchy, class for point, shape rectangle, square, circle, ellipse, triangle, polygenetic.

Program:-

```
class Shape {
   String shapeVar="This automatically tells the shape";
   int lengthOfSide1=0, lengthOfSide2=0;
   final double PI=3.14;

   void calcArea(){
      System.out.print("This method will calculate the area of the shape selected\n");
   }

   void setShape(String str){
      shapeVar=str;
   }

   void setSides(int side){
      lengthOfSide1=side;
   }

   void setSides(int side, int side2){
      lengthOfSide1=side;
      lengthOfSide2=side2;
   }

   public static void main(String[] args){
      System.out.println("\n\nAnmol Baranwal -- 2820208\n");
      Shape obj;
      obj=new rectangle();
      obj.calcArea();
   }

}

// =========  POINT  =========
class point extends Shape{

   point(){
      setShape("Point represents a dimensionless shape");
   }

   void calcArea(){
      System.out.print("It doesn't make sense to calculate area of point\n");
   }

   void getSides(){
```

```
        System.out.print("No of sides in point is : infinite\n");
    }
}

// =========  RECTANGLE  =========
class rectangle extends Shape{

    rectangle(){
        setShape("A rectangle is closed flat shape, having four sides, and each angle equal to 90
degrees. The opposite sides of the rectangle are equal and parallel.");
    }

    void calcArea(){
        setSides(6, 7);
        System.out.print("Area is: "+lengthOfSide1*lengthOfSide2);
    }

    void getSides(){
        System.out.print("No of sides in point is :"+4+"\n");
    }
}

// =========  SQUARE  =========
class square extends Shape{

    square(){
        setShape("Square is a plane figure with four equal sides and four right (90°) angles. A square is
a special kind of rectangle (an equilateral one) and a special kind of parallelogram (an equilateral and
equiangular one).");
    }

    void calcArea(){
        setSides(8);
        System.out.print("Area is: "+lengthOfSide1*lengthOfSide1);
    }

    void getSides(){
        System.out.print("No of sides in point is : "+4+"\n");
    }
}

// =========  CIRCLE  =========
class circle extends Shape{

    circle(){
        setShape("A circle is a round-shaped figure that has no corners or edges.");
    }

    void calcArea(){
        setSides(8);
```

```
        System.out.print("Area is: "+PI*lengthOfSide1*lengthOfSide1);
    }

    void getSides(){
        System.out.print("No of sides in point is : infinite\n");
    }
}

// =========  ELLIPSE  =========
class ellipse extends Shape{

    ellipse(){
        setShape("An ellipse is a circle that has been stretched in one direction, to give it the shape of
an oval.");
    }

    void calcArea(){
        System.out.print("I will not calculate the area of ellipse. It needs extreme hard work\n");
    }

    void getSides(){
        System.out.print("No of sides in point is : infinite\n");
    }
}

// =========  TRIANGLE  =========
class triangle extends Shape{

    triangle(){
        setShape("A triangle is a three-sided polygon, which has three vertices. The three sides are
connected with each other end to end at a point, which forms the angles of the triangle. The sum of
all three angles of the triangle is equal to 180 degrees.");
    }

    void calcArea(){
        setSides(8, 5);
        System.out.print("Area is: "+ 0.5*lengthOfSide1*lengthOfSide2);
    }

    void getSides(){
        System.out.print("No of sides in point is : "+3+"\n");
    }
}

// =========  POLYGENETIC  =========
class polygenetic extends Shape{

    polygenetic(){
        setShape("Polygenetic is similar to bell-shaped.");
    }
```

```
    void calcArea(){
        System.out.print("Do you know what you are asking me to do?\n");
    }

    void getSides(){
        System.out.print("No of sides in point is : infinite\n");
    }
}
```

OUTPUT:

```
Anmol Baranwal -- 2820208

Area is: 42
PS E:\coding\JavaLab\Shape class heirarchy>
```

Practical 4.
Design a simple test application to demonstrate dynamic polymorphism.

Program:-

```
class Person
{
   // private int childVar;
   int commonVar=20;

   void speak(){
      System.out.print("The parent is speaking\n");
   }

   public static void main(String[] args){
      System.out.println("\n\nAnmol Baranwal -- 2820208\n");

      Person obj;

      obj = new child1();
      obj.speak();

      obj = new child2();
      obj.speak();
      System.out.println(obj.commonVar);

   }

}

class child1 extends Person
{
   int commonVar=60;

   void speak(){
      System.out.print("The child1 is speaking\n");
   }

}

class child2 extends Person
{
   int commonVar=70;

   void speak(){
      System.out.print("The child2 is speaking\n");
   }
}
```

OUTPUT:

```
Anmol Baranwal -- 2820208

The child1 is speaking
The child2 is speaking
20
PS E:\coding\JavaLab\dynamicPolymorphism>
```

Practical 5.
Design a java interface for ADT Stack.

Program:-

```
import java.util.*;
import java.io.*;

interface InterfaceADTStack
{
   void push(int item);
   void pop();

}

class StackMethod implements InterfaceADTStack
{
   int stack[]= new int[5];
   int top=-1;
   int i;

   public void push(int item){
      if(top>=4)  System.out.println("Overflow Condition");

      top++;
      stack[top]= item;
      System.out.println("Item pushed: "+ stack[top]);
   }

   public void pop(){
      if(top<0)   System.out.println("underflow condition");

      System.out.println("item popped: "+stack[top]);
      top--;
   }

   public void displayStack(){
      if(top<0)   System.out.println("Stack is empty");

      for(int i=0;i<=top;i++){
         int temp=i+1;
         System.out.println("Element "+ temp + " is: "+stack[i]);
      }
   }
}
class TestStack
{
   public static void main(String[] args) throws IOException
   {
```

```
        System.out.println("\n\nAnmol Baranwal -- 2820208\n");

        StackMethod s= new StackMethod();
        int choice;
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("---------------------------------- STACK ----------------------------------");
        do{
           try{
              System.out.println("Type your choice for the appropriate function (1/2/3) ");
              System.out.println("1. push operation\n2. pop operation\n3. Display the stack\n4. Exit\n");
              choice= Integer.parseInt(in.readLine());
              switch(choice){
                 case 1:
                    System.out.println("Enter the value to include in stack");
                    int el= Integer.parseInt(in.readLine());
                    s.push(el);
                    break;
                 case 2:
                    s.pop();
                    break;
                 case 3:
                    System.out.println("The elements are: ");
                    s.displayStack();
                    break;
                 case 4:
                    System.out.println("You have reached a rare condition");
                    break;
              }
           }
           catch(IOException e){
              System.out.println("OOPS!! I/O Error");
           }
           System.out.println("Do u want to continue?? Press 1 to continue ");
           choice=Integer.parseInt(in.readLine());

        }while(choice==1);
     }
  }
```

OUTPUT:

```
Anmol Baranwal -- 2820208

---------------------------------- STACK ----------------------------------
Type your choice for the appropriate function (1/2/3)
1. push operation
2. pop operation
3. Display the stack
4. Exit

1
Enter the value to include in stack
2
Item pushed: 2
Do u want to continue?? Press 1 to continue
1
Type your choice for the appropriate function (1/2/3)
1. push operation
2. pop operation
3. Display the stack
4. Exit

1
Enter the value to include in stack
5
Item pushed: 5
Do u want to continue?? Press 1 to continue
1
Type your choice for the appropriate function (1/2/3)
1. push operation
2. pop operation
3. Display the stack
4. Exit

3
The elements are:
Element 1 is: 2
Element 2 is: 5
Do u want to continue?? Press 1 to continue
1
Type your choice for the appropriate function (1/2/3)
1. push operation
2. pop operation
3. Display the stack
4. Exit

2
item popped: 5
Do u want to continue?? Press 1 to continue
1
Type your choice for the appropriate function (1/2/3)
1. push operation
2. pop operation
3. Display the stack
4. Exit

2
item popped: 2
Do u want to continue?? Press 1 to continue
4
PS E:\coding\JavaLab\ADT Interface>
```

Practical 6.

Develop two different classes that implement this interface. One using array and other using linked list.

Program:-

```
import java.io.*;

interface Mystack
{
  public void pop();
  public void push();
  public void display();
}
class StackArray implements Mystack
{
  final static int n=5;
  int stack[]= new int[n];
  int top= -1;
  public void push(){
    try{
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      if(top==(n-1)){
        System.out.println("Stack Overflow\n");
        return;
      } else {
        System.out.print("Enter the element:  ");
        int ele=Integer.parseInt(br.readLine());
        stack[++top]=ele;
      }
    }
    catch(IOException e){
      System.out.println("e");
    }
  }

  public void pop(){
    if(top<0){
      System.out.println("Stack underflow");
      return;
    } else {
      int popper=stack[top];
      top--;
      System.out.println("Popped element: "+popper);
    }
  }

  public void display(){
    if(top<0){
```

```
      System.out.println("Stack is empty");
      return;
    } else {
      String str=" ";
      for(int i=0; i<=top; i++){
        if(i==top){
          str=str+" "+stack[i];
        }
        else  str=str+" "+stack[i]+" <--";

      }
      System.out.println("Elements are:"+str);
    }
  }
}

class Link
{
  public int data;
  public Link nextLink;
  public Link(int d){
    data= d;
    nextLink=null;
  }
  public void printLink(){
    System.out.print(" --> "+data);
  }
}

class StackLL implements Mystack
{
  private Link first; public StackLL(){
    first = null;
  }
  public boolean isEmpty(){
    return first == null;
  }
  public void push(){
    try{
      BufferedReader br= new BufferedReader(new InputStreamReader(System.in));
      System.out.print("Enter the element:  ");
      int ele=Integer.parseInt(br.readLine());
      Link link = new Link(ele);
      link.nextLink = first;
      first = link;
    }
    catch(IOException e){
      System.err.println(e);
    }
  }
```

```
  public Link delete(){
    Link temp =first;
    try{
      first = first.nextLink;
    }
    catch(NullPointerException e){
      throw e;
    }
    return temp;
  }

  public void pop(){
    try{
      Link deletedLink = delete();
      System.out.println("\nPopped: "+deletedLink.data);
    }
    catch(NullPointerException e){
      throw e;
    }
  }

  public void display(){
    if(first==null) System.out.println("\nStack is empty\n ");
    else{
      Link currentLink = first;
      System.out.print("\nElements are: ");
      while(currentLink != null){
        currentLink.printLink();
        currentLink =currentLink.nextLink;
      }
      System.out.println("");
    }
  }
}

class StackUsingArrayAndLL
{
  public static void main(String arg[])throws IOException
  {
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    System.out.println("\nAnmol Baranwal -- 2820208\n\n");
    System.out.println("==================================");
    System.out.println("Implementation of Stack using Array");
    System.out.println("==================================");
    StackArray st=new StackArray();
    int ch=0;
    do{
      System.out.println("\n1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List \n");
      System.out.print("Enter your choice:  ");
```

```
    ch=Integer.parseInt(br.readLine());
    switch(ch)
    {
      case 1:
        st.push();
        break;
      case 2:
        st.pop();
        break;
      case 3:
        st.display();
        break;
      case 4:
        System.out.println("Terminating Code..\n");
        return;
    }
   } while(ch<5);

   System.out.println("\n\n=======================================");
   System.out.println("Implementation of Stack using Linked List");
   System.out.println("=======================================");
   StackLL st1=new StackLL();
   ch=0;
   do{
    System.out.println("\n1.Push | 2.Pop | 3.Display | 4.Exit\n");
    System.out.print("Enter your choice:  ");
    ch=Integer.parseInt(br.readLine());
    switch(ch)
    {
      case 1:
        st1.push();
        break;
      case 2:
        try{
          st1.pop();
        }
        catch(NullPointerException e){
          System.out.println("Stack underflow");
        }
        break;
      case 3:
        st1.display();
        break;
      default:
        System.out.println("Terminating Code..\n");
        return;
    }
   } while(ch<5);
  }
}
```

OUTPUT:

```
Anmol Baranwal -- 2820208


==================================
Implementation of Stack using Array
==================================

1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  2
Stack underflow

1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  1
Enter the element:  4

1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  1
Enter the element:  7

1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  1
Enter the element:  2

1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  1
Enter the element:  13
1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  1
Enter the element:  6

1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  1
Stack Overflow


1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  2
Popped element: 6

1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  3
Elements are:  4 <-- 7 <-- 2 <-- 13

1.Push | 2.Pop | 3.Display | 4.Exit | 5.Use Linked List

Enter your choice:  5
```

```
========================================
Implementation of Stack using Linked List
========================================

1.Push | 2.Pop | 3.Display | 4.Exit

Enter your choice:  2
Stack underflow

1.Push | 2.Pop | 3.Display | 4.Exit

Enter your choice:  1
Enter the element:  5

1.Push | 2.Pop | 3.Display | 4.Exit

Enter your choice:  1
Enter the element:  9

1.Push | 2.Pop | 3.Display | 4.Exit

Enter your choice:  1
Enter the element:  11

1.Push | 2.Pop | 3.Display | 4.Exit

Enter your choice:  1
Enter the element:  8
1.Push | 2.Pop | 3.Display | 4.Exit

Enter your choice:  2

Popped: 8

1.Push | 2.Pop | 3.Display | 4.Exit

Enter your choice:  3

Elements are:  --> 11 --> 9 --> 5

1.Push | 2.Pop | 3.Display | 4.Exit

Enter your choice:  4
Terminating Code..
```

Practical 7.
Develop a simple paint like program that can draw basic graphical primitives.

Program:-

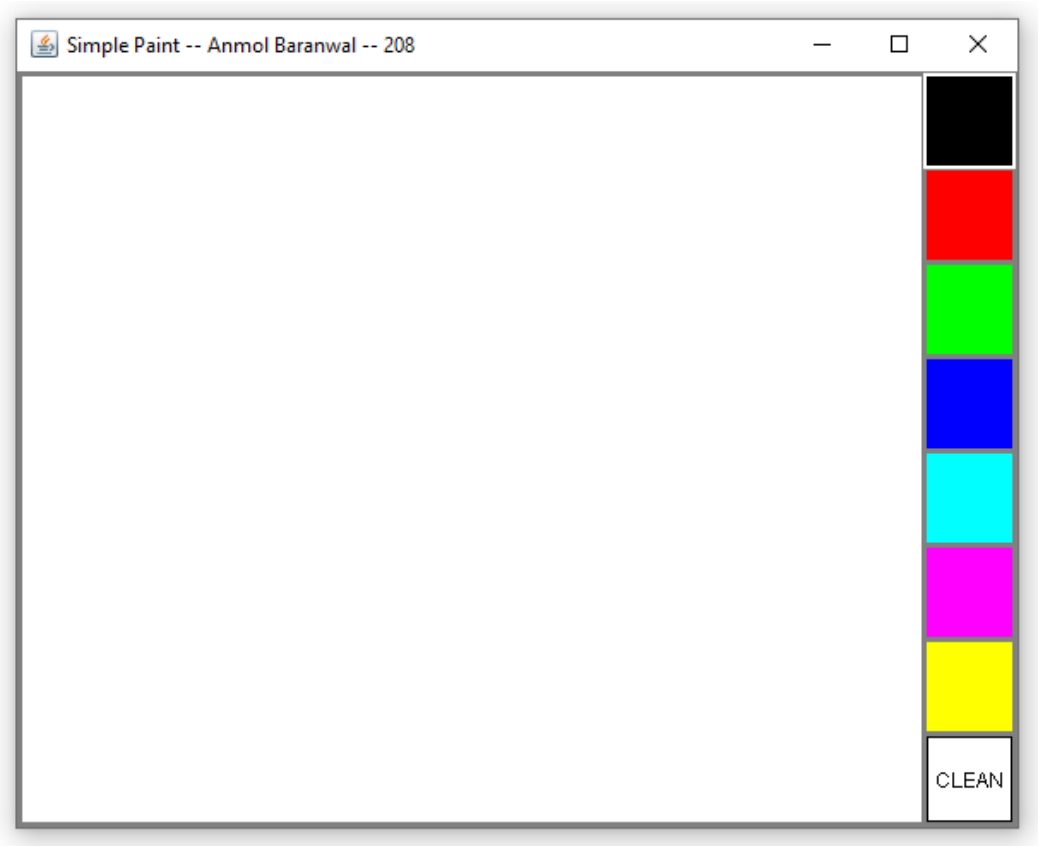
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class SimplePaint extends JApplet {

   public static void main(String[] args) {
      JFrame window = new JFrame("Simple Paint -- Anmol Baranwal -- 208");
      SimplePaintPanel content = new SimplePaintPanel();
      window.setContentPane(content);
      window.setSize(600, 480);
      window.setLocation(100, 100);
      window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      window.setVisible(true);
   }
   public void init() {
      setContentPane(new SimplePaintPanel());
   }

   public static class SimplePaintPanel extends JPanel implements MouseListener,
MouseMotionListener {

      private final static int BLACK = 0, RED = 1, GREEN = 2, BLUE = 3, CYAN = 4, MAGENTA
= 5, YELLOW = 6;

      private int currentColor = BLACK; // The current selected drawing color,

      private int prevX, prevY; // The previous location of the mouse.

      private boolean dragging; // true when the user is drawing.

      private Graphics graphicsForDrawing;

      SimplePaintPanel() {
         setBackground(Color.WHITE);
         addMouseListener(this);
         addMouseMotionListener(this);
      }

      public void paintComponent(Graphics g) {

         super.paintComponent(g); // Fill with background color (white)
         int width = getWidth(); // Width of the panel
         int height = getHeight(); // Height of the panel
```

```
    int colorSpacing = (height - 56) / 7;

    g.setColor(Color.GRAY);
    g.drawRect(0, 0, width - 1, height - 1);
    g.drawRect(1, 1, width - 3, height - 3);
    g.drawRect(2, 2, width - 5, height - 5);
    g.fillRect(width - 56, 0, 56, height);

    g.setColor(Color.WHITE);
    g.fillRect(width - 53, height - 53, 50, 50);
    g.setColor(Color.BLACK);
    g.drawRect(width - 53, height - 53, 49, 49);
    g.drawString("CLEAN", width - 48, height - 23);

    /* Drawing the seven color rectangles */
    g.setColor(Color.BLACK);
    g.fillRect(width - 53, 3 + 0 * colorSpacing, 50, colorSpacing - 3);
    g.setColor(Color.RED);
    g.fillRect(width - 53, 3 + 1 * colorSpacing, 50, colorSpacing - 3);
    g.setColor(Color.GREEN);
    g.fillRect(width - 53, 3 + 2 * colorSpacing, 50, colorSpacing - 3);
    g.setColor(Color.BLUE);
    g.fillRect(width - 53, 3 + 3 * colorSpacing, 50, colorSpacing - 3);
    g.setColor(Color.CYAN);
    g.fillRect(width - 53, 3 + 4 * colorSpacing, 50, colorSpacing - 3);
    g.setColor(Color.MAGENTA);
    g.fillRect(width - 53, 3 + 5 * colorSpacing, 50, colorSpacing - 3);
    g.setColor(Color.YELLOW);
    g.fillRect(width - 53, 3 + 6 * colorSpacing, 50, colorSpacing - 3);

    /* Drawing a 2-pixel white border around the rect
     of the current drawing color */
    g.setColor(Color.WHITE);
    g.drawRect(width - 55, 1 + currentColor * colorSpacing, 53, colorSpacing);
    g.drawRect(width - 54, 2 + currentColor * colorSpacing, 51, colorSpacing - 2);

}
private void changeColor(int y) {

    int width = getWidth(); // Width of applet
    int height = getHeight();
    int colorSpacing = (height - 56) / 7; // Space for one color rectangle
    int newColor = y / colorSpacing; // to get the color no that was clicked

    if (newColor < 0 || newColor > 6)   return; // check validity of color number

    Graphics g = getGraphics();
    g.setColor(Color.GRAY);
    g.drawRect(width - 55, 1 + currentColor * colorSpacing, 53, colorSpacing);
```

```
    g.drawRect(width - 54, 2 + currentColor * colorSpacing, 51, colorSpacing - 2);
    currentColor = newColor;
    g.setColor(Color.WHITE);
    g.drawRect(width - 55, 1 + currentColor * colorSpacing, 53, colorSpacing);
    g.drawRect(width - 54, 2 + currentColor * colorSpacing, 51, colorSpacing - 2);
    g.dispose();

}

private void setUpDrawingGraphics() {
    graphicsForDrawing = getGraphics();
    switch (currentColor) {
        case BLACK:
            graphicsForDrawing.setColor(Color.BLACK);
            break;
        case RED:
            graphicsForDrawing.setColor(Color.RED);
            break;
        case GREEN:
            graphicsForDrawing.setColor(Color.GREEN);
            break;
        case BLUE:
            graphicsForDrawing.setColor(Color.BLUE);
            break;
        case CYAN:
            graphicsForDrawing.setColor(Color.CYAN);
            break;
        case MAGENTA:
            graphicsForDrawing.setColor(Color.MAGENTA);
            break;
        case YELLOW:
            graphicsForDrawing.setColor(Color.YELLOW);
            break;
    }
}

public void mousePressed(MouseEvent evt) {

    int x = evt.getX(); // x-coordinate of user click
    int y = evt.getY(); // y-coordinate of user click

    int width = getWidth();
    int height = getHeight();

    //  Ignore mouse presses that occur when user is already drawing a curve.
    if (dragging == true)   return;

    if (x > width - 53) {
        if (y > height - 53)    repaint(); //  Clicked on "CLEAR button".
        else    changeColor(y); // Clicked on the color palette.
```

```
        } else if (x > 3 && x < width - 56 && y > 3 && y < height - 3) {
            // The user has clicked on the white drawing area.
            // Start drawing a curve from the point (x,y).
            prevX = x;
            prevY = y;
            dragging = true;
            setUpDrawingGraphics();
        }
    } // end mousePressed()

    public void mouseReleased(MouseEvent evt) {
        if (dragging == false)
            return;        // Nothing to do because the user isn't drawing.
        dragging = false;
        graphicsForDrawing.dispose();
        graphicsForDrawing = null;
    }

    public void mouseDragged(MouseEvent evt) {

        if (dragging == false)
            return;

        int x = evt.getX();
        int y = evt.getY();

        if (x < 3)      //  Adjust the value of x,
            x = 3;      //  to make sure it's in the drawing area.
        if (x > getWidth() - 57)
            x = getWidth() - 57;

        if (y < 3)
            y = 3;
        if (y > getHeight() - 4)
            y = getHeight() - 4;

        graphicsForDrawing.drawLine(prevX, prevY, x, y); // Draw the line.

        prevX = x;
        prevY = y;

    } // end mouseDragged()
    public void mouseEntered(MouseEvent evt) {}  // Some empty routines.
    public void mouseExited(MouseEvent evt) {}   // (Required by the MouseListener
    public void mouseClicked(MouseEvent evt) {}  // and MouseMotionListener
    public void mouseMoved(MouseEvent evt) {}    // interfaces).
  }
}
```

OUTPUT:


Simple Paint -- Anmol Baranwal -- 208
CLEAN


Simple Paint -- Anmol Baranwal -- 208
CLEAN

Practical 8.
Develop a scientific calculator using event driven programming.

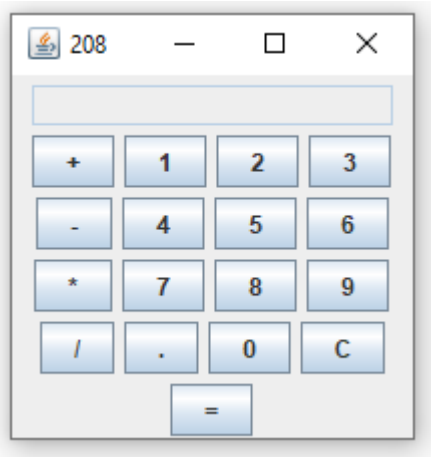
Program:-

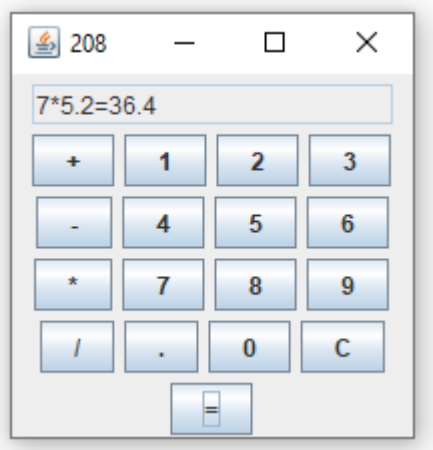
```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class JFrameCalculator extends JFrame implements ActionListener
{
   static JFrame f;
   static JTextField txtField;

   String s0, s1, s2;
   // creating JButton
   static JButton btn0, btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9, btnEQ, btnADD, btnMIN,
   btnDIV, btnMUL, btnC, btnDOT;

   // JPanel
   static JPanel p;

   public JFrameCalculator(){
      s0 = s1 = s2 = "";
   }

   public static void main(String args[]){
      f= new JFrame();
      p= new JPanel();

      JFrameCalculator jc = new JFrameCalculator();
      txtField= new JTextField(16);
      txtField.setEditable(false);

      // Creating a new buttons
      btn0= new JButton("0");
      btn1= new JButton("1");
      btn2= new JButton("2");
      btn3= new JButton("3");
      btn4= new JButton("4");
      btn5= new JButton("5");
      btn6= new JButton("6");
      btn7= new JButton("7");
      btn8= new JButton("8");
      btn9= new JButton("9");
      btnEQ= new JButton("=");
      btnADD= new JButton("+");
      btnMIN= new JButton("-");
      btnMUL= new JButton("*");
```

```
    btnDIV= new JButton("/");
    btnC= new JButton("C");
    btnDOT= new JButton(".");

    // add action listeners
    btn0.addActionListener(jc);
    btn1.addActionListener(jc);
    btn2.addActionListener(jc);
    btn3.addActionListener(jc);
    btn4.addActionListener(jc);
    btn5.addActionListener(jc);
    btn6.addActionListener(jc);
    btn7.addActionListener(jc);
    btn8.addActionListener(jc);
    btn9.addActionListener(jc);
    btnEQ.addActionListener(jc);
    btnADD.addActionListener(jc);
    btnMIN.addActionListener(jc);
    btnMUL.addActionListener(jc);
    btnDIV.addActionListener(jc);
    btnC.addActionListener(jc);
    btnDOT.addActionListener(jc);

    // add buttons to panel
    p.add(txtField);
    p.add(btnADD);
    p.add(btn1);
    p.add(btn2);
    p.add(btn3);
    p.add(btnMIN);
    p.add(btn4);
    p.add(btn5);
    p.add(btn6);
    p.add(btnMUL);
    p.add(btn7);
    p.add(btn8);
    p.add(btn9);
    p.add(btnDIV);
    p.add(btnDOT);
    p.add(btn0);
    p.add(btnC);
    p.add(btnEQ);

    f.add(p);

    f.setBounds(0, 0, 200, 220);
    f.setTitle("208");
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setVisible(true);
}
```

```
  public void actionPerformed(ActionEvent e)
 {
  String s = e.getActionCommand();
  // if the value is a number
  if ((s.charAt(0) >= '0' && s.charAt(0) <= '9') || s.charAt(0) == '.') {
   // if operand is present then add to second no
   if (!s1.equals("")) s2 = s2 + s;
   else    s0 = s0 + s;

   txtField.setText(s0 + s1 + s2);
  }
  else if (s.charAt(0) == 'C') {
   s0 = s1 = s2 = "";
   txtField.setText(s0 + s1 + s2);
  }
  else if (s.charAt(0) == '=') {

   double te;
   // store the value in 1st
   if (s1.equals("+")) te = (Double.parseDouble(s0) + Double.parseDouble(s2));
   else if (s1.equals("-"))    te = (Double.parseDouble(s0) - Double.parseDouble(s2));
   else if (s1.equals("/"))    te = (Double.parseDouble(s0) / Double.parseDouble(s2));
   else    te = (Double.parseDouble(s0) * Double.parseDouble(s2));

   txtField.setText(s0 + s1 + s2 + "=" + te);

   // convert it to string
   s0 = Double.toString(te);
   s1 = s2 = "";
  }
  else {
   if (s1.equals("") || s2.equals("")) s1 = s;  // no operand
   else {
    double te;
    if (s1.equals("+")) te = (Double.parseDouble(s0) + Double.parseDouble(s2));
    else if (s1.equals("-"))    te = (Double.parseDouble(s0) - Double.parseDouble(s2));
    else if (s1.equals("/"))    te = (Double.parseDouble(s0) / Double.parseDouble(s2));
    else    te = (Double.parseDouble(s0) * Double.parseDouble(s2));

    s0 = Double.toString(te);
    s1 = s;
    s2 = "";
   }
   txtField.setText(s0 + s1 + s2);
  }
 }
}
```

OUTPUT:





Practical 9.
Develop a template for linked list class along with its members in Java.

Program:-

```
import java.io.*;
class Node{
   int data;
   Node next;

   Node(int d){
      data= d;
      next= null;
   }
}
public class LinkedList
{
   Node head;
   public static LinkedList insert(LinkedList ll, int data){
      Node newNode= new Node(data);
      newNode.next= null;

      if (ll.head == null)  ll.head = newNode;
      else {
         Node last = ll.head;
         while (last.next != null) {
            last = last.next;
         }
         last.next = newNode;
      }
      return ll;
   }
   public static void printList(LinkedList ll)
   {
      Node currNode = ll.head;
      System.out.println("LinkedList: ");

      while(currNode != null){
         System.out.print(currNode.data + " ");
         currNode = currNode.next;
      }
   }
   public static void main(String args[]){
      System.out.println("\n\nAnmol Baranwal – 2820208\n");
      LinkedList ll = new LinkedList();
      for(int i=1; i<9; i++)  insert(ll, i);  // ll= insert(ll, 1); without loop
      printList(ll);
   }
}
```

OUTPUT:

```
Anmol Baranwal -- 2820208

LinkedList:
1 2 3 4 5 6 7 8
PS E:\coding\JavaLab\Linked List Java>
```

Practical 10.
Write a program to insert and view data using Servlets.

Program:-

In Servlet.java

```
Import java.io.IOException;
 Import java.io.PrintWriter;
 Import java.sql.*;
Import javax.servlet.ServletException;
 Import javax.servlet.annotation.WebServlet;
 Import javax.servlet.http.HttpServlet;
Import javax.servlet.http.HttpServletRequest;
 Import javax.servlet.http.HttpServletResponse;

@WebServlet(“/MovieServlet”)
Public class MovieServlet extends HttpServlet
 { Private static final long serialVersionUID = 1L;
Public MovieServlet() { Super();
}
Protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
Response.getWriter().append(“Served at: “).append(request.getContextPath());
}
Protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
PrintWriter pw;
Response.setContentType(“text/html”); Pw=response.getWriter();

String name=request.getParameter(“name”);
 String actor=request.getParameter(“actor”);
String actress=request.getParameter(“actress”);
String director=request.getParameter(“director”);
String rDate=request.getParameter(“rDate”);
String rPoint=request.getParameter(“rPoint”); Float rating=Float.parseFloat(rPoint);




Try
{

Class.forName(“com.mysql.jdbc.Driver”);
String url=”jdbc:mysql:
//localhost:3306/bcapracticals”;
String user=”root”;
String password=”admin”;
```

```
Connection con=DriverManager.getConnection(url, user, password);
String query=”Insert into movie(name,actor,actress,director,releaseDate,ratingPoint) values
(?,?,?,?,?,?);
PreparedStatement pstmt=con.prepareStatement(query);
Pstmt.setString(1, name);
mt.setString(2, actor);
Pstmt.setString(3,actress);
 Pstmt.setString(4, director);
 Pstmt.setString(5, rDate);
 Pstmt.setFloat(6,rating);
Int x=pstmt.executeUpdate();
 If(x==1)
{
Pw.println(“Values Inserted Successfully”);
}
}
Catch(Exception e)
{
 e.printStackTrace();
  }
Pw.close();
}
```

HTML Form

```
<!DOCTYPE html>
<html>
<head>
   <meta charset=”UTF-8”>
   <title>Insert title here</title>
</head>
<body>
   <form method=”Post” action=”/BCAPracticals/MovieServlet”> <div>
      <label>Enter Movie Name: </label>
      <input type="text" name="name">
      </div>
      <div>
         <label>Enter Actor Name: </label>
         <input type="text" name="actor"></div>
      <div>
         <label>Enter Actress Name: </label>
         <input type="text" name="actress">
      </div>
      <div>
         <label>Enter Director Name: </label>
         <input type="text" name="director">
      </div>
      <div>
         <label>Enter Release Date: </label>
```

```
            <input type="text" name="rDate">
        </div>
        <div><label>Enter Rating Point: </label>
            <input type="text" name="rPoint">
        </div>
        <div>
            <button type="submit" value="Add-movie">Submit</button>
        </div>
    </form>
</body>
</html>
```

OUTPUT:

Enter Movie Name:
Enter Actor Name:
Enter Actress Name:
Enter Director Name:
Enter Release Date:
Enter Rating Point:
Submit