Lab Manual with Solutions

1. ***Write a Java Applications to extract a portion of a character string and print the extracted string.***

```
import java.util.Scanner;
/*
 @Purpose : Find the substring of a string given the start and end index numbers.
 */
public class ExtractString {
  public static void main(String args[]) {
     String string, subString;
     int firstIndex, secondIndex;
     Scanner in = new Scanner(System.in);
     System.out.println("Enter the String : ");
     string = in.next();
     System.out.println("Enter the index 1 : ");
     firstIndex = in.nextInt();
     System.out.println("Enter the index 2 : ");
     secondIndex = in.nextInt();
     subString = string.substring(firstIndex, secondIndex);
     System.out.println("The Substring is : " + subString);
  }
}
```

OUTPUT :

```
Enter the String :
BeautifulDay
Enter the index 1 :
9
Enter the index 2 :
12
The Substring is : Day
```

2. ***Write a Java Program to implement the concept of multiple inheritance using Interfaces***

```
/*
@Purpose : Implement multiple interface using Interfaces.
 */
class Student {
  int rollNo;
  String name = "Smith Jones";

  void setRollNumber(int rollNumber) {
     rollNo = rollNumber;
  }

  void printStudentDetails() {
     System.out.println("RollNo   :  " + rollNo);
     System.out.println("Name     :  " + name);
  }
}

class StudentTest extends Student {
  int mark1, mark2;

  void setMarks(int firstMark, int secondMark) {
     mark1 = firstMark;
     mark2 = secondMark;
  }

  public void printMarks() {
     System.out.println("Mark1    :  " + mark1);
     System.out.println("Mark2    :  " + mark2);
  }
}

interface Sports {
  int sportsMark = 75;

  void printSportsMark();
}

class Result extends StudentTest implements Sports {
  int totalMarks;

  public void printSportsMark() {
     System.out.println("SportMark=" + sportsMark);
  }

  void displayTotalMarks() {
     totalMarks = mark1 + mark2 + sportsMark;
     System.out.println("\tSCORE CARD");
     printStudentDetails();
     printMarks();
     System.out.println("\n\nSports Marks   :" + sportsMark);
     System.out.println("Total Marks    :" + totalMarks);
  }
}

class ExampleMultipleInheritance {
  public static void main(String args[]) {
     Result studentObject = new Result();
     studentObject.setRollNumber(1000);
     studentObject.setMarks(75, 100);
     studentObject.displayTotalMarks();
  }
}
```

```
OUTPUT :

  SCORE CARD

RollNo   :  1000
Name     :  Smith Jones
Mark1    :  75
Mark2    :  100
Sports Marks   :75



Total Marks    :250
```

3. ***Write a Java Program to create an Exception called payout-of-bounds and throw the exception.***

```
import java.io.DataInputStream;
import java.io.IOException;
/*
@Purpose : This program throws an exception when the payout is out of bounds.
*/
public class ExceptionCheck {
  public static void main(String args[]) throws IOException {
```

```
  int payAmount;
  DataInputStream inputAmount = new DataInputStream(System.in);
  System.out.println("\n\nCHECK PAY MODULE");
  System.out.println("* * *");
  System.out.println("\n Enter a Basic Pay Amount   : ");
  payAmount = Integer.parseInt(inputAmount.readLine()); // This denotes the deprecated function
                                                       // DataInputStream Class.

  /* A piece of code enclosed within a try - catch block will be handled by the
     Exception Handler */
  try {
     if (payAmount > 1000)
     throw new PayoutOfBoundException("Basic Pay is Out of Bound");
     else
       System.out.println("\n Given Basic Pay is:" + payAmount);
     } catch (Exception exception) {
        System.out.println("Caught:" + exception);
     }
  }
}
class PayoutOfBoundException extends IOException {
  private static final long serialVersionUID = 1L;
  PayoutOfBoundException(String message) {
     System.out.println("\nOOPS!!!! -----   "+message);
  }
}
```

OUTPUT :

```
CHECK PAY MODULE
* * *

 Enter a Basic Pay Amount   :
1200

OOPS!!!! -----   Basic Pay is Out of Bound
Caught:PayoutOfBoundException
```

4. ***Write a Java Program to implement the concept of multithreading with the use of any three multiplication tables and assign three different priorities to them.***

```
import java.lang.Thread;
/*
 @Purpose : The purpose of this program is to implement multithreading using 3 threads
 */
class FirstThread extends Thread {
  public void run() {
   int firstNumber = 0, secondNumber = 2, result;
   for (firstNumber = 1; firstNumber <= 4; firstNumber++) {
    result = firstNumber * secondNumber;
    System.out.println("From thread FirstThread:"+firstNumber+"*"+secondNumber+"="+result);
    }
    System.out.println("====== \n Exit from FirstThread \n====== \n ");
  }
}

class SecondThread extends Thread {
  public void run() {
    int firstNumber, secondNumber = 3, result;
    for (firstNumber = 1; firstNumber <= 4; firstNumber++) {
       result = firstNumber * secondNumber;
       System.out.println("From thread SecondThread:"+firstNumber+"*"+secondNumber+"="+result);
    }
    System.out.println("====== \n Exit from SecondThread \n====== \n ");
  }
}

class ThirdThread extends Thread {
  public void run() {
    int firstNumber, secondNumber = 5, result;
    for (firstNumber = 1; firstNumber <= 4; firstNumber++) {
       result = firstNumber * secondNumber;
       System.out.println("From ThirdThread:"+firstNumber+"*"+firstNumber+"="+result);
    }
    System.out.println("====== \n Exit from ThirdThread \n====== \n ");
  }
}

class ExampleMultiThreading {
  public static void main(String args[]) {
    int tempPriority = 0;
```

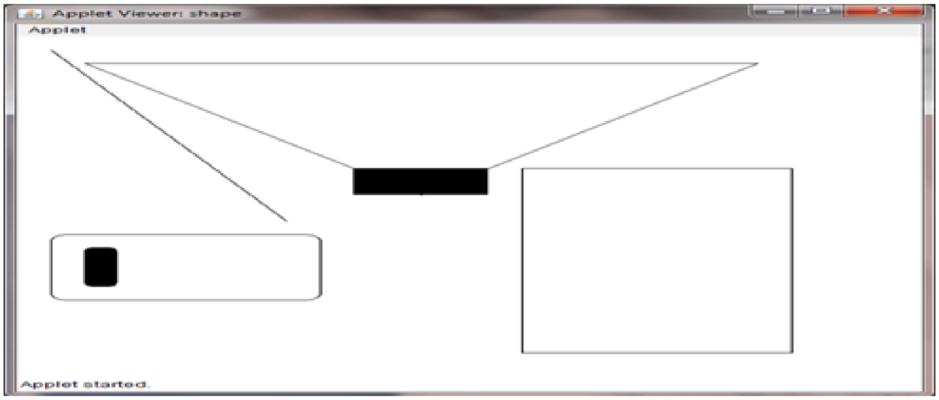
```
        FirstThread firstThreadObj = new FirstThread();
        SecondThread secondThreadObj = new SecondThread();
        ThirdThread thirdThreadObj = new ThirdThread();
        firstThreadObj.setPriority(Thread.MAX_PRIORITY);
        secondThreadObj.setPriority(firstThreadObj.getPriority() + tempPriority);
        thirdThreadObj.setPriority(Thread.MIN_PRIORITY);
        secondThreadObj.setPriority(firstThreadObj.getPriority() + tempPriority);
        thirdThreadObj.setPriority(Thread.MIN_PRIORITY);
        System.out.println("----\nStart thread FirstThread\n----");
        firstThreadObj.start();
        System.out.println("----\nStart thread SecondThread\n----");
        secondThreadObj.start();
        System.out.println("----\nStart thread ThirdThread\n----");
        thirdThreadObj.start();
    }
}
```

```
OUTPUT :

----
Start thread FirstThread
----
----
Start thread SecondThread
----
----
Start thread ThirdThread
----
From thread FirstThread:1*2=2
From thread FirstThread:2*2=4
From thread FirstThread:3*2=6
From thread SecondThread: 1*3=3
From thread FirstThread:4*2=8
From thread SecondThread: 2*3=6
======
 Exit from FirstThread
======

From thread SecondThread: 3*3=9
From thread SecondThread: 4*3=12
======
 Exit from SecondThread
======

From ThirdThread :1*1=5
From ThirdThread :2*2=10
From ThirdThread :3*3=15
From ThirdThread :4*4=20
======
 Exit from ThirdThread
======
```

5. *Write a Java Program to draw several shapes in the created windows.*

```
import java.awt.*;
import java.applet.*;
/*
 @Purpose : This applet draws several shapes in the window created.
 */
public class Shapes extends Applet {
  private static final long serialVersionUID = 1L;
  int xPoint[] = { 60, 240, 440, 40 };
  int yPoint[] = { 60, 240, 40, 40 };
  int nPoints = 4;

  public void paint(Graphics g) {
     g.drawPolygon(xPoint, yPoint, nPoints);
     g.drawLine(20, 20, 160, 280);
     g.drawRect(300, 200, 160, 280);
     g.fillRect(200, 200, 80, 40);
     g.fillRoundRect(40, 320, 20, 60, 10, 10);
     g.drawRoundRect(20, 300, 160, 100, 20, 20);
     g.drawOval(550, 400, 100, 40);
  }
}
/*
 * <html> <head> <applet code="shapes",height=400 width=400> </applet> </head>
 * </html>
 */
```

OUTPUT:

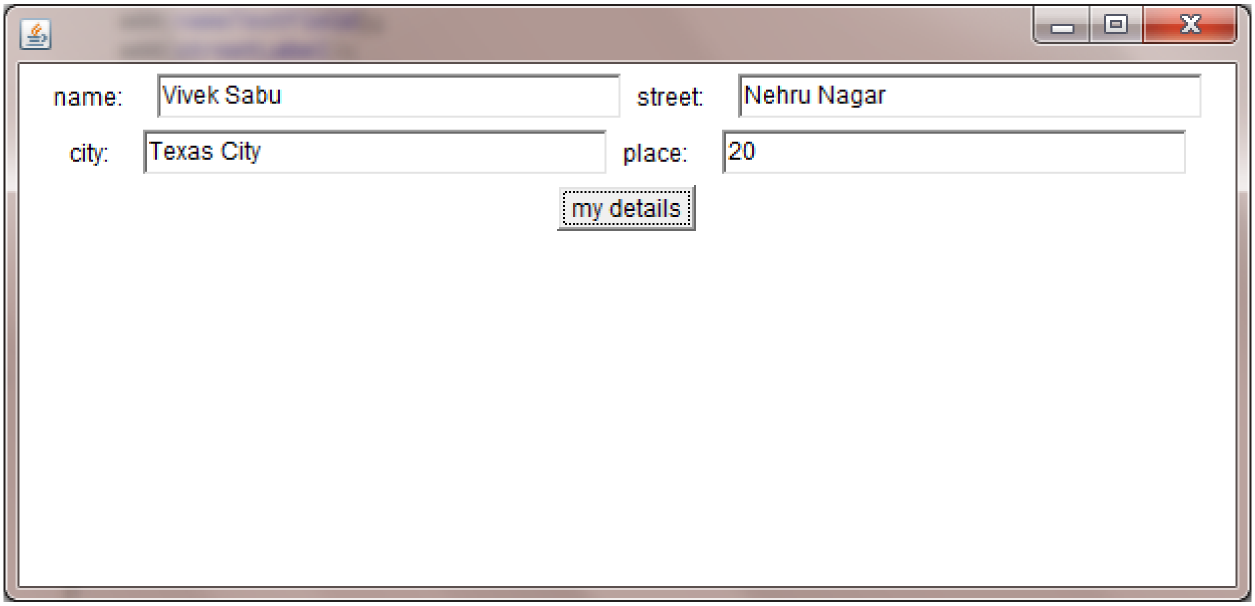


6. *Write a Java Program to create a frame with four text fields name, street, city and pin code with suitable tables. Also add a button called "my details". When the button is clicked its corresponding values are to be appeared in the text fields*

```
import java.awt.*;
import java.awt.event.*;
/*
@Purpose : This Java Program creates a frame with four text fields name, street, city and pin
code with suitable tables.   Also add a button called "my details"
When the button is clicked its corresponding values are to be appeared in the text fields
 */
public class ButtonClick extends Frame implements ActionListener {
  private static final long serialVersionUID = 1L;
  TextField nameTextField, streetTextField, cityTextField, placeTextField;
  Label nameLabel, streetLabel, cityLabel, placeLabel;
  Button button1;

  ButtonClick() {
     setLayout(new GridLayout(4, 2));
     nameTextField = new TextField(30);
     streetTextField = new TextField(30);
     cityTextField = new TextField(30);
     placeTextField = new TextField(30);
     nameLabel = new Label("name:", Label.LEFT);
     streetLabel = new Label("street:", Label.LEFT);
     cityLabel = new Label("city:", Label.LEFT);
     placeLabel = new Label("place:", Label.LEFT);
     button1 = new Button("my details");
     add(nameLabel);
     add(nameTextField);
     add(streetLabel);
```

```
    add(streetTextField);
    add(cityLabel);
    add(cityTextField);
    add(placeLabel);
    add(placeTextField);
    setLayout(new FlowLayout(FlowLayout.CENTER));
    button1.addActionListener(this);
    add(button1);
  }

  public void actionPerformed(ActionEvent actionEventObj) {
    if (actionEventObj.getSource() == button1) {
      nameTextField.setText("Vivek Sabu");
      streetTextField.setText("Nehru Nagar");
      cityTextField.setText("Texas City");
      placeTextField.setText("20");
    }
  }

  public static void main(String arg[]) {
    ButtonClick buttonObj = new ButtonClick();
    buttonObj.setSize(600, 300);
    buttonObj.addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
        System.exit(0);
      }
    });
    buttonObj.show();
  }
}
```

OUTPUT:



name: Vivek Sabu street: Nehru Nagar

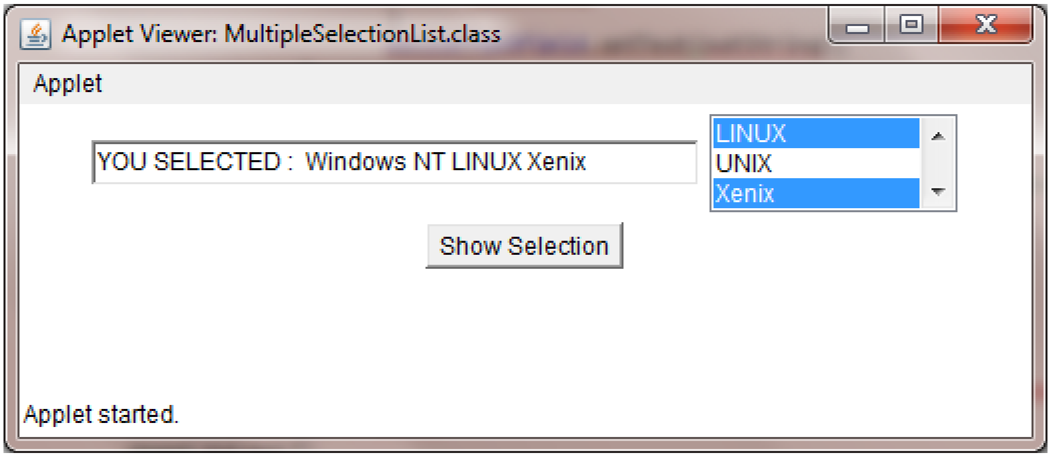
city: Texas City place: 20

7. *Write a Java Program to demonstrate the Multiple Selection List-box.*

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
/*
 @Purpose : This Java applet creates multiple selection list box.
 */
public class MultipleSelectionList extends Applet implements ActionListener {
  private static final long serialVersionUID = 1L;
  List osList;
  TextField osListTextField;
  Button showButton;
  String Selections[];

  public void init() {
     osListTextField = new TextField(40);
     add(osListTextField);
     osList = new List(3, true);
     osList.add("Windows NT");
     osList.add("Windows Vista");
     osList.add("Windows XP");
     osList.add("LINUX");
     osList.add("UNIX");
     osList.add("Xenix");
     add(osList);
     showButton = new Button("Show Selection");
```

```
      showButton.addActionListener(this);
      add(showButton);
   }

   public void actionPerformed(ActionEvent e) {
      String outString = new String("YOU SELECTED : ");
      if (e.getSource() == showButton) {
         osListTextField.setText("");
         Selections = osList.getSelectedItems();
         for (int loopIndex=0;loopIndex<Selections.length;loopIndex++) {
            outString += " " + Selections[loopIndex];
         }
         osListTextField.setText(outString);
      }
   }
}
/*
 * <html> <applet code = MultipleSelectionList.class width=400 height=400>
 * </applet> </html>
 */
```

OUTPUT:



YOU SELECTED : Windows NT LINUX Xenix

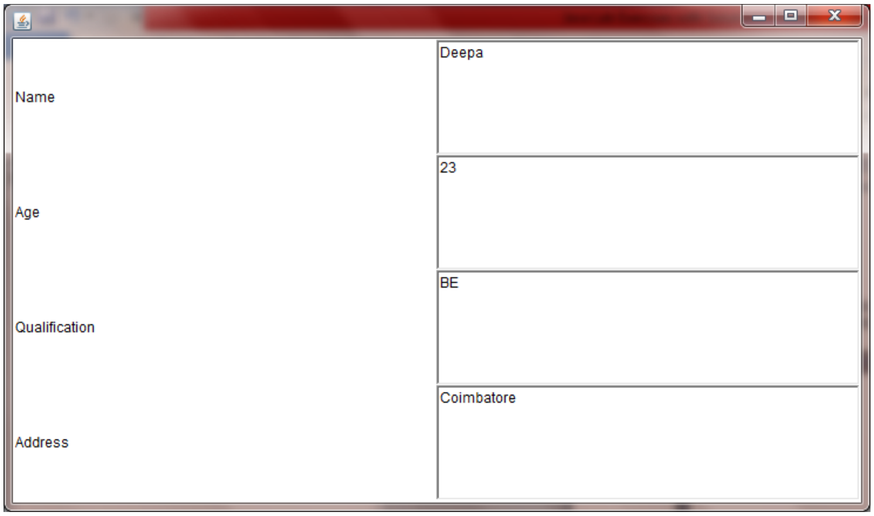
Applet started.

8. ***Write a Java Program to create a frame with three text fields for name, age and qualification and a text field for multiple line for address***

```
import java.awt.*;
import java.awt.event.*;
/*
@Purpose : This Program creates a frame with three text fields for name, age, qualification and
a text field for multiple line for address
*/
```

```
public class PersonalDetails extends Frame {
  private static final long serialVersionUID = 1L;
  TextField nameText, ageText, qualificationText, addressText;
  Label nameLabel, ageLabel, qualificationLabel, addressLabel;
  TextArea textArea;

  PersonalDetails() {
    setLayout(new GridLayout(4, 2));
    nameText = new TextField(30);
    ageText = new TextField(30);
    qualificationText = new TextField(30);
    addressText = new TextField(40);
    textArea = new TextArea(" ", 2, 5);
    nameLabel = new Label("Name", Label.LEFT);
    ageLabel = new Label("Age", Label.LEFT);
    qualificationLabel = new Label("Qualification", Label.LEFT);
    addressLabel = new Label("Address", Label.LEFT);
    add(nameLabel);
    add(nameText);
    add(ageLabel);
    add(ageText);
    add(qualificationLabel);
    add(qualificationText);
    add(addressLabel);
    add(addressText);
  }

  public static void main(String[] args) {
    PersonalDetails personalDetails = new PersonalDetails();
    personalDetails.setSize(300, 250);
    personalDetails.addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
        System.exit(0);
      }
    });
    personalDetails.show();
  }
}
```

OUTPUT:



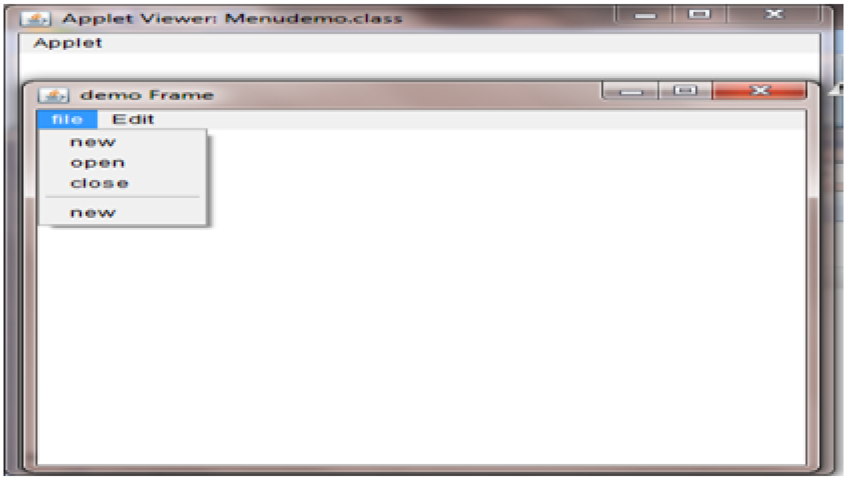
Name	Deepa
Age	23
Qualification	BE
Address	Coimbatore

9. *Write a Java Program to create Menu Bars and pull down menus*

```
import java.awt.*;
import java.applet.*;
/*
@Purpose : This Applet creates a menu bar and sets menu items on the menu bar.
 */
public class MenuBarApplication extends Applet {
  private static final long serialVersionUID = 1L;

  public void init() {
     /* method initializationOfMenuBar is described. */
     int width = Integer.parseInt(getParameter("width"));
     int height = Integer.parseInt(getParameter("height"));
     Frame frame = new Frame("First Frame");
     /* Frame constructor with String Parameter   */
     frame.resize(width, height);
     MenuBar menuBar = new MenuBar();
     frame.setMenuBar(menuBar);
     Menu menuFile = new Menu("File");/* Menu constructor with String Parameter */
     menuFile.add(new MenuItem("New"));
     menuFile.add(new MenuItem("Open"));
     menuFile.add(new MenuItem("Close")); /* File operation on Menu */
     menuFile.add(new MenuItem("-"));
     menuFile.add(new MenuItem("new"));
     menuBar.add(menuFile);
     Menu menuEdit = new Menu("Edit");/* Menu constructor with String Parameter */
     menuEdit.add(new MenuItem("Cut"));
     menuEdit.add(new MenuItem("Copy")); /* Menu Items to edit */
```

```
    menuEdit.add(new MenuItem("paste"));
    menuEdit.add(new MenuItem("-"));
    Menu menu = new Menu("Special");/* Menu constructor with String Parameter */
    menu.add(new MenuItem("First"));
    menu.add(new MenuItem("Second")); /* Add Menu Items */
    menu.add(new MenuItem("Third"));
    menuEdit.add(menu);
    menuEdit.add(new CheckboxMenuItem("Debug"));
    menuEdit.add(new CheckboxMenuItem("Testing"));
    menuBar.add(menuEdit);
    frame.show();
  }
}
/*
 * TO RUN THE APPLET <html> <head> <applet code="Menudemo.class" width=400
 * height=400> </applet> </head> </html>
 */
```

OUTPUT:



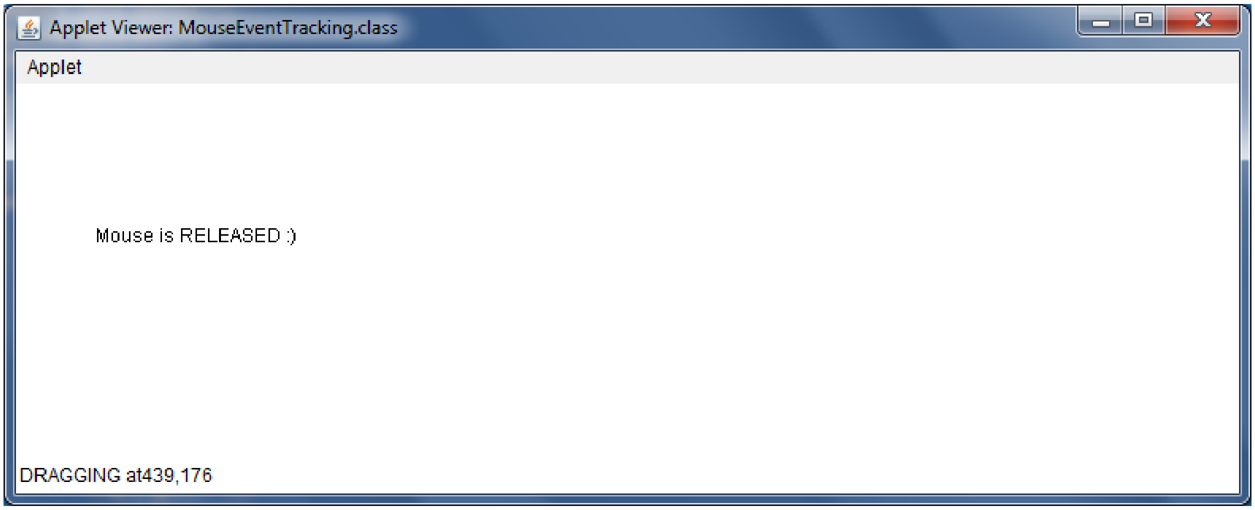
10. Write a Java Program to create frames which respond to the mouse clicks. For each events with mouse such as mouse up, mouse down, etc., the corresponding message to be displayed.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*
 @Purpose : This Java Applet tracks the Mouse Events and displays appropriate messages.
 */
public class MouseEventTracking extends Applet implements MouseListener, MouseMotionListener {
  private static final long serialVersionUID = 1L;
  String msg = " ";
```

```
   int msgheight, msgwidth, mousex = 0, mousey = 0;

   public void init() {
      addMouseListener(this);
      addMouseMotionListener(this);
   }

   public void mousePressed(MouseEvent me) {
      msgheight = 50;
      msgwidth = 100;
      msg = "Mouse Pressed";
      repaint();
   }

   public void mouseClicked(MouseEvent me) {
      msgheight = 50;
      msgwidth = 100;
      msg = "You CLICKED your Mouse";
      repaint();
   }

   public void mouseExited(MouseEvent me) {
      msgheight = 50;
      msgwidth = 100;
      msg = "Oops! You have EXITED";
      repaint();
   }

   public void mouseReleased(MouseEvent me) {
      msgheight = 50;
      msgwidth = 100;
      msg = "Mouse is RELEASED :)";
      repaint();
   }

   public void mouseEntered(MouseEvent me) {
      msgheight = 50;
      msgwidth = 100;
      msg = "Mouse ENTERING the frame!!!";
      repaint();
```

```
    }

    public void mouseDragged(MouseEvent me) {
       msg = "";
       showStatus("DRAGGING at" + me.getX() + "," + me.getY());
       repaint();
    }

    public void mouseMoved(MouseEvent me) {
       showStatus("MOVED at" + me.getX() + "," + me.getY());
       repaint();
    }

    public void paint(Graphics g) {
       g.drawString(msg, msgheight, msgwidth);
    }
  }
  /*
   * TO RUN THE APPLET
   * <html> <head> <applet code="MouseEvents", width=400
   * height=400> </applet> </head> </html>
   */
```

OUTPUT:



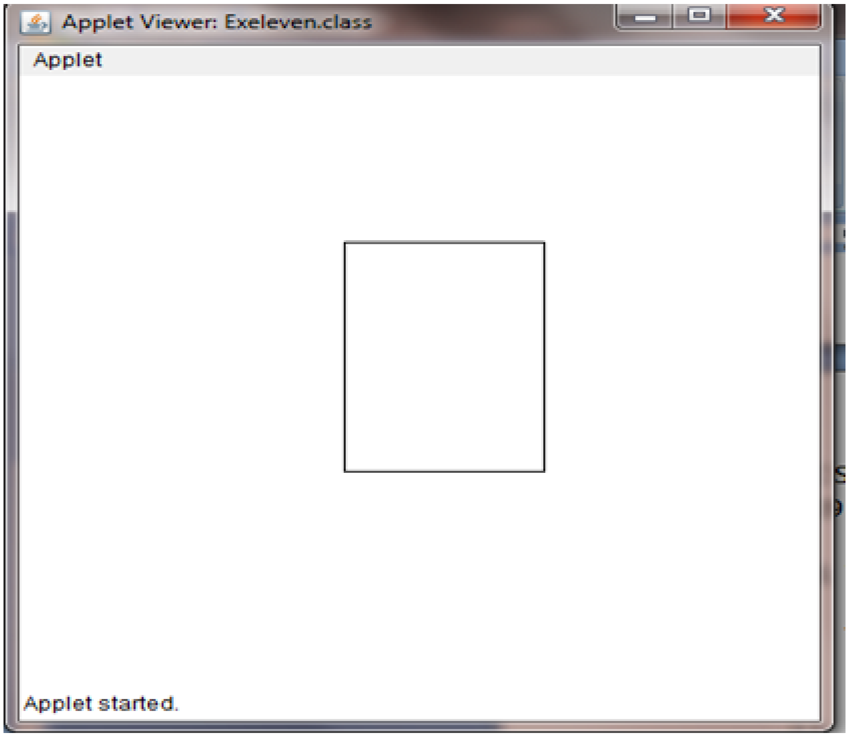
```
Mouse is RELEASED :)
DRAGGING at439,176
```

11. Write a Java Program to draw circle, square, ellipse and rectangle at the mouse click positions.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
```

```
/*
 @Purpose : This Applet draws a shape on the screen.
 */
public class MouseClick extends Applet {
  private static final long serialVersionUID = 1L;
  int length = 0, width = 0, count = 0;

  public void init() {
    addMouseListener(new MouseAdapter() {
      public void mousePressed(MouseEvent event) {
        count++;
        length = event.getX();
        width = event.getY();
        repaint();
      }
    });
  }

  public void Paint(Graphics g) {
    if (count % 4 == 1)
      g.drawOval(length, width, 150, 150);
    else if (count % 4 == 2)
      g.drawRect(length, width, 150, 150);
    else if (count % 4 == 3)
      g.drawOval(length, width, 100, 150);
    else if (count % 4 == 0)
      g.drawRect(length, width, 100, 150);
  }
}
/*
 * <html> <head> <title>Mouse click</title></head> <body> <applet code =
 * Mouseclick.class width=400 height=400> </applet> </body> </html>
 */
```

OUTPUT:



12. Write a Java Program which open an existing file and append text to that file.

```
import java.io.*;
/*
 @Purpose : This Program opens an existing file and appends text(CHENNAI) to that file.
 */
public class RandomAccessFileTryOut {
  public static void main(String args[]) {
     RandomAccessFile randomFile;
     try {
        randomFile = new RandomAccessFile("city.txt", "rw");
        randomFile.seek(randomFile.length());
        randomFile.writeBytes("\nCHENNAI");
        randomFile.close();
     } catch (IOException ioe) {
        System.out.println(ioe);
     }
  }
}
```

OUTPUT:

```
CHENNAI
```