Q1. 1.1 Develop a program to select multiple languages known to user. (e.g Marathi. Hindi, English, Sanskrit)

```
import java.applet.Applet;
import java.awt.*;
public class Pract1 extends Applet {
    List cities;
   public void init() {
      this.cities =new List(4);
      this.cities.add("marathi");
      this.cities.add("Hindi");
      this.cities.add("English");
      this.cities.add("Sanskrit");
      add(cities);
     }
}
```

1.2 Write a program to create three buttons with caption OK , RESET ,CANCEL.

```
import java.applet.Applet;
import java.awt.*;
public class Pract2 extends Applet {
   public void init()
   {
      Button btn1 = new Button("OK");
      Button btn2 = new Button("RESET");
      Button btn3 = new Button("CANCEL");
      add(btn1);
      add(btn2);
      add(btn3);
   }
  }
```

Q2. 2.1 Develop an applet using List components to add names of 10 different cities.

```
package pract3;
import java.applet.Applet;
import java.awt.*;
public class Pract3 extends Applet {
  List cities;
   public void init() {
      this.cities =new List(10);
      this.cities.add("mumbai");
      this.cities.add("pune");
      this.cities.add("nashik");
      this.cities.add("nagpur");
      this.cities.add("sangli");
      this.cities.add("kolhapur");
      this.cities.add("satara");
      this.cities.add("karad");
      this.cities.add("jalna");
      this.cities.add("amramati");
      this.add(this.cities);

   }
}
```

2.2 Develop applet to select multiple names of news papers.

```
package pract4;
import java.applet.Applet;
import java.awt.*;
public class Pract4 extends Applet {
  List n;
   public void init() {
      this.n =new List(10);
      this.n.add("Times of India");
      this.n.add("Pudhari");
      this.n.add("Maharashtra Times");
      this.n.add("Punya Nagri");
      this.n.add("The Hindu");
     this.add(this.n);
 } }
```

Q3. Write a program for making a calculator of 4*4 Grid.

```
package pract5;
import java.applet.Applet;
import java.awt.*;
public class Pract5 extends Applet {
  public void init() {
      setLayout(new GridLayout(4, 4));
      for (int i = 0; i <= 9; i++) {
         Button button = new Button(String.valueOf(i));
         add(button);
      }
      Button addButton = new Button("+");
      add(addButton);
 Button subtractButton = new Button("-");
      add(subtractButton);
```

```
        Button multiplyButton = new Button("*");
        add(multiplyButton);
        Button divideButton = new Button("/");
        add(divideButton);
    }
   }
```

Q4. Develop a java program to create a 5 buttons using Grid Layout

package pract6;

```
import java.applet.Applet;
import java.awt.*;
public class Pract6 extends Applet{
    public void init() {
        setLayout(new GridLayout(5,5));
        for(int i=1;i<=25;i++){
            Button bb = new Button(String.valueOf(i));
            add(bb);
        } } }
```

Q5. Write a program to develop a frame to select the different states of India using JcomboBox.

```
import javax.swing.*;
public class Pract8 {

Pract8(){
  JFrame f=new JFrame();
    String s[]={"Maharashtra","Punjab","Gujrat","TamilNadu"};
    JComboBox cb=new JComboBox(s);
    cb.setBounds(90, 50,150,20);
    f.add(cb);
    f.setLayout(null);
    f.setSize(400,400);
    f.setVisible(true);
```

```
}
public static void main(String[] args) {
   new Pract8();
}
}
```

Q6. Write a JTree program to show root directory and its sub-folder of your system

```
import javax.swing.JFrame;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
public class Pract7 extends JFrame {
   private JTree tree;
   public Pract7() {
      DefaultMutableTreeNode root = new DefaultMutableTreeNode("This PC");
      DefaultMutableTreeNode f = new DefaultMutableTreeNode("Windows :C");
      DefaultMutableTreeNode n= new DefaultMutableTreeNode("New Volume :D");
      root.add(f);
      root.add(n);
     DefaultMutableTreeNode a = new DefaultMutableTreeNode("eclipse");
      DefaultMutableTreeNode b = new DefaultMutableTreeNode("Intel");
      DefaultMutableTreeNode c = new DefaultMutableTreeNode("oneschool-master");
      DefaultMutableTreeNode d = new DefaultMutableTreeNode("PerfLogs");
      DefaultMutableTreeNode e = new DefaultMutableTreeNode("Program File");
      f.add(a);
      f.add(b);
      f.add(c);
      n.add(d);
      n.add(e);
     tree = new JTree(root);
      add(tree);
     setTitle("Language Tree");
      setSize(300, 300);
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setVisible(true);
    }
        public static void main(String[] args) {
        new Pract7();
    }
}
```

Q7.Write a java program to create a table of name of student ,Percentage and Grade of 10 students using JTable

```
import java.awt.BorderLayout;
import javax.swing.JApplet;
import javax.swing.JTable;
import javax.swing.ScrollPaneConstants;
import javax.swing.JScrollPane;
public class pract9 extends JApplet
{
    public void init()
    {
        setVisible(true);
        setSize(400,400);
        //setLayout( new BorderLayout() );
        String collumnHeading[] = {"Name","Percentage","Grade"};
        Object data[][]={
            {"A1",98,"A"},
            {"A2",90,"C"},
            {"A3",88,"A"},
            {"A4",99,"A"},
            {"A5",59,"A"},
            {"A6",94,"D"},
```

```
        {"A7",92,"A"},
        {"A8",42,"C"},
        {"A9",85,"A"},
        {"A10",98,"B"}
      };

      JTable JTableObj = new JTable(data,collumnHeading);

      int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
      int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
      JScrollPane jsp = new JScrollPane(JTableObj,v,h);

      add(jsp,BorderLayout.CENTER);
   }
}
```

Q8.Write a program using JProgressBar to show the progress of progressbar when user clicks on JButton

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class pract10 extends JApplet implements ActionListener
{
   JProgressBar JProgressBarObj;
   JButton JButtonObj;
   int i=0;

   public void init()
   {
      setSize(400,400);
      setVisible(true);
```

```
        setLayout(new FlowLayout());
        JButtonObj = new JButton("Click Me");
        JButtonObj.addActionListener(this);
        JProgressBarObj = new JProgressBar();
        JProgressBarObj.setStringPainted(true);
        JProgressBarObj.setValue(0);
        add(JButtonObj);
        add(JProgressBarObj);
    }
    public void actionPerformed(ActionEvent ie)
    {
        this.iterate();
    }
    public void iterate()
    {
        while(i<=2000)
        {
            JProgressBarObj.setValue(i);
            i=i+20;
            try
            {
                Thread.sleep(10);
            }
            catch(Exception e)
            {}
}} }
```

Q9.Write a program to generate KeyEvent when a key is pressed and display "key pressed " message.

```
import java.awt.*;
import java.awt.event.*;
public class pract11 extends Frame implements KeyListener {
public pract11() {
        addKeyListener(this);
    }
    public void keyPressed(KeyEvent e) {
        System.out.println("Key pressed: " + e.getKeyChar());
    }
  public void keyReleased(KeyEvent e) {}
    public void keyTyped(KeyEvent e) {  }
public static void main(String[] args) {
        pract11 example = new pract11();
        example.setSize(300, 300);
        example.setVisible(true);
    }
}
```

Q10. Develop a program to accept two numbers and display product of two numbers when user pressed "Multiply" button.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
class pract12 extends JFrame implements ActionListener                //implement listener interface
{
 JLabel l1, l2;
 JTextField t1, t2, t3;
 JButton b1;
 public pract12()
```

```
{
 l1 = new JLabel("First Number:");
 l1.setBounds(20, 10, 100, 20);               //x, y, width, height
 t1 = new JTextField(10);
 t1.setBounds(120, 10, 100, 20);
 l2 = new JLabel("Second Number:");
 l2.setBounds(20, 40, 100, 20);
 t2 = new JTextField(10);
 t2.setBounds(120, 40, 100, 20);
 b1 = new JButton("Product");
 b1.setBounds(20, 70, 80, 20);
 t3 = new JTextField(10);
 t3.setBounds(120, 70, 100, 120);
 add(l1); add(t1);  add(l2);  add(t2); add(b1);  add(t3);
 b1.addActionListener(this);      //Registering event
 setSize(400,300);
 setLayout(null);
 setVisible(true);
 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
 @Override
 public void actionPerformed(ActionEvent e)      {
   if(e.getSource()==b1){
       int num1 = Integer.parseInt(t1.getText());
       int num2 = Integer.parseInt(t2.getText());
       int product = num1 * num2;
       t3.setText(String.valueOf(product));
  }
 public static void main(String args[]){
   new pract12();
 } }
```

Q11. Write the program to count the number of clicks performed by the user in a Frame window.

```
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JFrame;

public class pract13 {
   private static int clickCount = 0;

   public static void main(String[] args) {
      JFrame frame = new JFrame("Click Counter");
      frame.setSize(300, 200);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      frame.addMouseListener(new MouseAdapter() {
         @Override
         public void mouseClicked(MouseEvent e) {
            clickCount++;
            System.out.println("Click count: " + clickCount);
         }
      });

      frame.setVisible(true);
   }
}
```

Q12. Write the program to demonstrate the use of mouseDragged and mouseMoved method of MouseMotionListener

```
import java.awt.event.MouseMotionAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseMotionListener;
import java.applet.Applet;

public class pract14 extends Applet
{
   public void init()
   {
      addMouseMotionListener(new MouseDrag(this));
   }
}
class MouseDrag extends MouseMotionAdapter
{
   pract14  ad;
   public MouseDrag(pract14  ad)
   {
      this.ad = ad;
   }
   public void mouseDragged(MouseEvent me)
   {
      ad.showStatus("Mouse Dragged");
   }

}
```

Q13.Write a program using JPasswordField and JTextField to demonstrate the use of user Authentication

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class pract15 extends JFrame implements ActionListener {
   JLabel userLabel, passLabel, messageLabel;
   JTextField userTextField;
   JPasswordField passTextField;
   JButton loginButton, resetButton;
   pract15 () {
      setTitle("User Authentication");
      userLabel = new JLabel("Username:");
      passLabel = new JLabel("Password:");
      userTextField = new JTextField();
      passTextField = new JPasswordField();
      loginButton = new JButton("Login");
      resetButton = new JButton("Reset");
      messageLabel = new JLabel();
      setLayout(null);
      userLabel.setBounds(50, 70, 100, 30);
      passLabel.setBounds(50, 110, 100, 30);
      userTextField.setBounds(150, 70, 150, 30);
      passTextField.setBounds(150, 110, 150, 30);
      loginButton.setBounds(50, 160, 100, 30);
      resetButton.setBounds(200, 160, 100, 30);
      messageLabel.setBounds(50, 200, 250, 30);
      add(userLabel);  add(passLabel);  add(userTextField);  add(passTextField);
      add(loginButton);   add(resetButton);      add(messageLabel);
      loginButton.addActionListener(this);
      resetButton.addActionListener(this);
```

```
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == loginButton) {
            String userText;
            String passText;
            userText = userTextField.getText();
            passText = new String(passTextField.getPassword());
            if (userText.equalsIgnoreCase("admin") && passText.equalsIgnoreCase("password")) {
                messageLabel.setForeground(Color.green);
                messageLabel.setText("Login Successful");
            } else {
                messageLabel.setForeground(Color.red);
                messageLabel.setText("Invalid Username or Password");
            }
        } else if (e.getSource() == resetButton) {
            userTextField.setText("");
            passTextField.setText("");
            messageLabel.setText("");
        }
    }

    public static void main(String[] args) {
        pract15  login = new pract15 ();
        login.setVisible(true);
        login.setSize(350, 300);
        login.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Q14. Develop a program using InetAddress class to retrieve IP address of computer when hostname is entered by the user.

```
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.util.Scanner;
public class {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter Host Name: ");
      String host = sc.nextLine();
      try{
         InetAddress ip = InetAddress.getByName(host);
         System.out.println("IP Adress of Computer is:"+ip.getHostAddress());
      }  catch(UnknownHostException e){
         System.out.print(e);
      }}}
```

Q15.Write a program using URL class to retrieve the host ,protocol,port and file of URL http://www.msbte.org.in.

```
import java.net.URL;
import java.net.MalformedURLException;
public class pract17{
   public static void main(String[] args) throws MalformedURLException  {
      URL url = new URL("https://msbte.org.in/");
      System.out.println("Authority: "+ url.getAuthority());
      System.out.println("Default Port: "+ url.getDefaultPort());
      System.out.println("File: "+ url.getFile());
      System.out.println("Path: "+ url.getPath());
      System.out.println("Protocol: "+ url.getProtocol());
      System.out.println("Reference: "+ url.getRef());
   }
}
```

Q16. Write a program using Socket and ServerSocket to create Chat Application.

```
import java.net.ServerSocket;
import java.net.Socket;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintStream;
import java.io.InputStreamReader;
public class pract18_ServerSide{
   public static void main(String[] args) throws IOException {
      ServerSocket s = new ServerSocket(2019);
      System.out.println("Server Started, waiting for client");
      Socket s1 = s.accept();
      BufferedReader br = new BufferedReader(
         new InputStreamReader(s1.getInputStream())
      );
      OutputStream out = s1.getOutputStream();
      PrintStream ps  = new PrintStream(out);

      BufferedReader br1 = new BufferedReader(
         new InputStreamReader(System.in)
      );do{
         String res = br.readLine();
         System.out.println("Client Send: "+res);
         System.out.print("Server: ");
         String msg = br1.readLine();
         System.out.print("\n\n");
         ps.println(msg);       }
      while(true);
   }
}
```

```
import java.net.Socket;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintStream;
public class pract18_ClientSide
{
   public static void main(String[] args) throws IOException {
      Socket s = new Socket("localhost",2545);
      System.out.println("Client Started, waiting for server response..");
      BufferedReader br = new BufferedReader(
         new InputStreamReader(System.in)
      );
      OutputStream os = s.getOutputStream();
      BufferedReader br1 = new BufferedReader(
         new InputStreamReader(s.getInputStream())
         );
      PrintStream ps = new PrintStream(os);
      do{
         System.out.print("Client: ");
         String msg = br.readLine();
         ps.println(msg);
         String res = br1.readLine();
         System.out.println("Server Send:"+res+"\n\n");
      }
      while(true);
   }
}
```