a)//student class

```
class Student
{
int hollticket;
String studName;
String Department;
Student(int ht, String sdN, String Dep)
{
hollticket=ht;
studName=sdN;
Department=Dep;
System.out.print("The student details are:"+hollticket+" "+studName+" "+Department+"\n");
}
}
class StudentDetails
{
public static void main(String args[])
{
Student obj1 = new Student(1,"Sasi","cse");
Student obj2 = new Student(2,"Bhanu","cse");
Student obj3 = new Student(3,"Anuhya","cse");
Student obj4 = new Student(4,"Abhinav","cse");
}
}
```

1b)//factorial

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code=Fact width=500 height=500> </applet>*/
public class Fact extends Applet implements ActionListener
```

```
{
Button b1,b2;
Label l1,l2;
TextField tf1,tf2;
public void init()
{
b1=new Button("COMPUTE");
b1.addActionListener(this);
b2=new Button("CLEAR");
b2.addActionListener(this);
tf1=new TextField(20);
tf2=new TextField(20);
l1=new Label("NUMBER");
l2=new Label("RESULT");
add(l1);
add(tf1);
add(l2);
add(tf2);
add(b1);
add(b2);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b1)
{
int a=Integer.parseInt(tf1.getText());
int fact=1;
```

```
for(int i=1;i<=a;i++)
fact*=i;
tf2.setText(""+fact);
}
else
{
tf1.setText("");
tf2.setText("");
} } }
```

Set-2

---

a)//overloading

```
class MethodOverLoad
{
void calValue()
{
int x=20;
x=x*x;
System.out.println("Sqrt of x is:"+x);
}
void calValue(int y)
{
y=y*y*y;
System.out.println("Cube of y is:"+y);
}
void calValue(int m,int n)
{
```

```
int z=m*n;
System.out.println("Product of m and n is:"+z);
} }
class MOL
{
public static void main(String args[])
{
MethodOverLoad m=new MethodOverLoad();
m.calValue();
m.calValue(10,20);
m.calValue(10);
}
}

//overriding
	class Human{
//Overridden method
public void eat()
{
System.out.println("Human is eating");
}
}


class Boy extends Human{
//Overriding method
public void eat(){
```

```
System.out.println("Boy is eating");
}
public static void main( String args[]) {
Boy obj = new Boy();
//This will call the child class version of eat()
obj.eat();
}
}
```

b)//traffic lights

```
 import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="Traffic" width=400 height=400>
</applet>*/
public class Traffic extends Applet implements ItemListener
{
int colourNum;
CheckboxGroup cbg;
Checkbox red,yellow,green;
String msg=" ";
public void init()
{
cbg=new CheckboxGroup();
red=new Checkbox("RED",cbg,true);
yellow=new Checkbox("YELLOW",cbg,true);
green=new Checkbox("GREEN",cbg,true);
```

```
add(red);
add(yellow);
add(green);
red.addItemListener(this);
yellow.addItemListener(this);
green.addItemListener(this);
}
public void itemStateChanged(ItemEvent ie)
{
if (ie.getSource()==red)
colourNum=1;
else if (ie.getSource()==yellow)
colourNum=2;
else
colourNum=3;
repaint();
}
public void paint (Graphics g)
{
g.setColor(Color.BLACK);
g.fillOval (150, 70, 50, 50); // red light
g.fillOval (150, 150, 50, 50); // yellow light
g.fillOval (150, 230, 50, 50); // green light
switch (colourNum)
{
case 1:g.setColor (Color.RED);
g.fillOval (150,70,50,50);
```

```
msg="STOP";
g.drawString(msg,210,100);
break;
case 2:g.setColor(Color.YELLOW);
g.fillOval (150,150,50,50);
g.setColor (Color.red);
msg="READY";
g.drawString(msg,210,180);
break;
case 3:g.setColor(Color.GREEN);
g.fillOval (150,230,50,50);
g.setColor (Color.red);
msg="GO";
g.drawString(msg,210,260);
break;
}
}
}
```

Set-3

---

a)
```
//primenumbers
import java.util.Scanner;
class a
{
public static void main(String[] args)
{
int n,p;
Scanner s=new Scanner(System.in);
System.out.println("Enter upto which number prime numbers are
needed");
```

```
n=s.nextInt();
System.out.println("<------------Prime numbers------------->");
for(int i=2;i<n;i++)
{
p=0;
for(int j=2;j<i/2;j++)
{
if(i%j==0)
p=1;
}
if(p==0)
System.out.println(i);
}
}
}
```

b)//synchronized

```
class Callme {
void call(String msg) {
 System.out.print("[" + msg);
try { Thread.sleep(1000);
 } catch(InterruptedException e) {
 System.out.println("Interrupted");
 } System.out.println("]");
} }
class Caller implements Runnable {
String msg; Callme target;
Thread t;
 public Caller(Callme targ, String s)
 { target = targ; msg = s;
 t = new Thread(this,"demo");
 t.start();
} public void run() {
synchronized(target){ target.call(msg);
```

```
}}}
class Synch {
 public static void main(String args[]) {
 Callme target = new Callme();
 Caller ob1 = new Caller(target, "Hello");
 Caller ob3 = new Caller(target, "World");
 }}
```

Set-4

---

a)//quadratic equation

```
   import java.util.Scanner;
class Roots
{
 public static void main(String args[])
 {
 int a,b,c;
 double x,y;
 Scanner s=new Scanner(System.in);
 System.out.println("Enter the values of a,b, and c");
 a=s.nextInt();
 b=s.nextInt();
 c=s.nextInt();
 double f=(b*b)-4*a*c;
System.out.println("F value="+f);
 if(f<0)
{
 System.out.println("No real roots");
 }
```

```
else
{
double l=Math.sqrt(f);
x=((-b-l)/(2*a));
y=((-b+l)/(2*a));
System.out.println("Roots of given equation:"+x+"\t"+y);
}
}
}
```

b)//keyevents

```
 import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
/* <APPLET CODE ="AppletKeyboard.class" WIDTH=300 HEIGHT=200>
</APPLET> */
public class AppletKeyboard extends Applet implements KeyListener
  {
      TextField t,tt,tp,tr;
       public void init()
          {
             t=new TextField(20);
             t.addKeyListener(this);
             tt=new TextField(70);
             tp=new TextField(70);
             tr=new TextField(70);
             add(t);
             add(tt);
             add(tr);
             add(tp);
          }
             public void keyTyped(KeyEvent e)
                 {
                    tt.setText("key Released"+e.getKeyChar());
                 }
                    public void keyReleased(KeyEvent e)
                      {
                         tr.setText("key Released"+e.getKeyChar());
                      }
                         public void keyPressed(KeyEvent e)
                           {
```

```
        int kc;
        String s;
        kc=e.getKeyCode();
        s=e.getKeyText(kc);
        tp.setText("Key Pressed"+s);
      }
  }
```

set-5

a)//voter

```
class InvalidAgeException extends Exception{
 InvalidAgeException(String s){
 super(s);
 }
}
class TestCustomException1{

 static void validate(int age)throws InvalidAgeException{
if(age<18)
 throw new InvalidAgeException("not valid");
 else
 System.out.println("welcome to vote");
 }

 public static void main(String args[]){
 try{
 validate(13);
 }catch(Exception m){System.out.println("Exception occured: "+m);}

 System.out.println("Custom exception Demo completed...");
```

```
 }
}
b)//mouse events
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="MouseEvents" width=300 height=300>
</applet>*/
public class MouseEvents extends Applet implements MouseListener,
MouseMotionListener
{
String msg = "";
int mouseX = 0, mouseY = 0; // coordinates of mouse
public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}
public void mouseClicked(MouseEvent me)
{
mouseX = 0;
mouseY = 10;
msg = "Mouse clicked.";
repaint();
}
public void mouseEntered(MouseEvent me)
{
```

```
mouseX = 0;
mouseY = 10;
msg = "Mouse entered.";
repaint();
}
public void mouseExited(MouseEvent me)
{
mouseX = 0;
mouseY = 10;
msg = "Mouse exited.";
repaint();
}
public void mousePressed(MouseEvent me)
{
mouseX = me.getX();
mouseY = me.getY();
msg = "Down";
repaint();
}
public void mouseReleased(MouseEvent me)
{
mouseX = me.getX();
mouseY = me.getY();
msg = "Up";
repaint();
}
public void mouseDragged(MouseEvent me)
```

```
{
mouseX = me.getX();
mouseY = me.getY();
msg = "*";
showStatus("Dragging mouse at " + mouseX + ", " + mouseY);
repaint();
}
public void mouseMoved(MouseEvent me)
{
showStatus("Moving mouse at " + me.getX() + ", " + me.getY());
}
public void paint(Graphics g)
{
g.drawString(msg, mouseX, mouseY);
}
}
```

Set-6

---

a)//sorting names

```
 import java.util.Scanner;
class SortStrings
{
 public static void main(String args[])
 {
 String temp;
 Scanner SC = new Scanner(System.in);
```

```
System.out.print("Enter the value of N: ");
int N= SC.nextInt();
SC.nextLine();

String names[] = new String[N];

System.out.println("Enter names: ");
for(int i=0; i<N; i++)
{
System.out.print("Enter name [ " + (i+1) +" ]: ");
names[i] = SC.nextLine();
}

//sorting strings
for(int i=0; i<5; i++)
{
for(int j=1; j<5; j++)
{
if(names[j-1].compareTo(names[j])>0)
{
temp=names[j-1];
names[j-1]=names[j];
names[j]=temp;
}
}
}
```

```
 System.out.println("\nSorted names are in Ascending Order: ");
 for(int i=0;i<N;i++)
 {
 System.out.println(names[i]);
 }
 }
}
```

b)//simple calculator

```
 import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="Calc" height=300 width=300>
</applet>
*/
public class Calc extends Applet implements ActionListener
{
TextField tf;
double arg=0;
String op="=";
boolean start=true;
public void init()
{
setLayout(new BorderLayout());
tf=new TextField("0");
add(tf,BorderLayout.NORTH);
```

```
Panel p=new Panel();
p.setLayout(new GridLayout(4,4));
String buttons="123/456*789-0.+=";
for(int i=0;i<buttons.length();i++)
{
Button b=new Button(buttons.substring(i,i+1));
p.add(b);
b.addActionListener(this);
}
add(p);
}
public void actionPerformed(ActionEvent ae)
{
String s=ae.getActionCommand();
if('0'<=s.charAt(0)&&s.charAt(0)<='9'||s.equals("."))
{
if(start)
tf.setText(s);
else
tf.setText(tf.getText()+s);
start=false;
}
else
{
calcu(Double.parseDouble(tf.getText()));
op=s;
start=true;
```

```
}
}
public void calcu(double n)
{
if(op.equals("+"))
arg+=n;
else
if(op.equals("-"))
arg-=n;
else
if(op.equals("*"))
arg*=n;
else
if(op.equals("/"))
{
try{
 arg/=n;
 }
 catch(ArithmeticException e)
 {
 tf.setText("Arithmetic Exception");
 }
}
else
if(op.equals("="))
arg=n;
tf.setText(""+arg);
```

}}

Set-7

```
 a)//frequency words
 package cmrit;
import java.util.Scanner;
import java.lang.String;
public class frequency_count {
 public static void main(String args[])
 {
 Scanner SC = new Scanner(System.in);
System.out.println("Enter the String: ");
 String str=SC.nextLine();
 System.out.println("Enter substring: ");
 String sub=SC.nextLine();
 int index,count=0;
 for(int i=0; i+sub.length()<=str.length(); i++)
 //i+sub.length() is used to reduce comparisions
 {
//System.out.println("i="+i);
//System.out.println("sub.length= "+sub.length()+ " " +
 // "i+sub.length()= "+ i+sub.length()+""
 // + "string.length ="+str.length());
 index=str.indexOf(sub,i);
 //System.out.println("index position of substring= "+index);
 if(index>=0)
 {
```

```
 count++;
 i=index;
 index=-1;
//System.out.println("count= "+count+" i= "+ ""+i+" ind= "+ind);
 }
 }
 System.out.println("Occurence of '"+sub+"' in String is "+
 count);
 }
}
```

b)//jscrollpanel

```
   import javax.swing.*;
 public class TabbedPaneExample {
 JFrame f;
 TabbedPaneExample(){
 f=new JFrame();
 JTextArea ta=new JTextArea(200,200);
 JPanel p1=new JPanel();
 p1.add(ta);
 JPanel p2=new JPanel();
 JPanel p3=new JPanel();
 JTabbedPane tp=new JTabbedPane();
 tp.setBounds(50,50,200,200);
tp.add("main",p1);
 tp.add("visit",p2);
 tp.add("help",p3);
```

```
f.add(tp);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public static void main(String[] args) {
new TabbedPaneExample();
} }

//jtable
import javax.swing.*;
public class TableEx
{
JFrame f;
TableEx()
{
f=new JFrame();
String data[][]={ {"501","ravi","50000"},
{"502","raju","70000"},
{"503","ramu","90000"}};
String column[]={"ID","NAME","SALARY"};
JTable jt=new JTable(data,column);
JScrollPane sp=new JScrollPane(jt);
f.add(sp);
f.setSize(300,400);
f.setVisible(true);
```

```
 f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }
 public static void main(String[] args)
 {
 new TableEx();
 }
 }
```

Set-8

---

a)//static

```
  class JavaExample{
 static int num;
 static String mystr;
 static{
 num = 97;
 mystr = "Static keyword in Java";
 }

 public static void main(String args[])
 {
 System.out.println("Value of num: "+num);
 System.out.println("Value of mystr: "+mystr);
 }
}

 //final
class Bike9{
```

```
 final int speedlimit=90;//final variable
 void run(){
 int speedlimit=400;
 System.out.println(speedlimit);
 }
 }
 class finalvar1{

 public static void main(String args[]){
 Bike9 obj=new Bike9();
 obj.speedlimit=100;
//obj.run();
 }
 }
```

b)//multi-thread

```
    import java.util.*;
class Odd implements Runnable
{
int i;
Odd(int i)
{
this.i = i;
}
public void run()
{
System.out.println("Cube of "+i+" is "+(i*i*i));
```

```
}
}
class Even implements Runnable
{
int i;
Even(int i)
{
this.i = i;
}
public void run()
{
System.out.println("Sqaure of "+i+" is "+(i*i));
}
}
class Random1 extends Thread
{
public void run()
{
int j=0;
Random rand = new Random();
while(true){
j = rand.nextInt(20);
if(j%2!=0)
{
Odd o = new Odd(j);
Thread t1 = new Thread(o);
t1.start();
```

```
try
{
Thread.sleep(2000);
}
catch(InterruptedException ie)
{ }
}
else
{
Even e = new Even(j);
Thread t2 = new Thread(e);
t2.start();
try
{
Thread.sleep(1000);
}
catch(InterruptedException ie)
{ }
}
}
}
}
public class mt6c
{
public static void main(String[] a)
{
Random1 r = new Random1();
```

```
Thread t = new Thread(r);
t.start();
}
}
```

Set-9

```
a)//multi inheritance
   interface Printable{
void print();
}
interface Showable{
void show();
}
class multiple_inheritance implements Printable, Showable{
public void print(){System.out.println("Hello");}
public void show(){System.out.println("Welcome");}

public static void main(String args[]){
multiple_inheritance obj = new multiple_inheritance ();
obj.print();
obj.show(); }
}

b)//division
  import java.applet.*;
  import java.awt.*;
  import java.awt.event.*;
```

```
import javax.swing.*;
/*<applet code=Div width=500 height=500>
</applet>*/
public class Div extends Applet implements ActionListener
{
Button b1,b2;
Label l1,l2,l3;
 TextField tf1,tf2,tf3;
String msg;
public void init()
{
 b1=new Button("COMPUTE");
 b1.addActionListener(this);
 b2=new Button("CLEAR");
 b2.addActionListener(this);
 tf1=new TextField(20);
 tf2=new TextField(20);
 tf3=new TextField(20);
 l1=new Label("NUMBER1");
l2=new Label("NUMBER2");
 l3=new Label("RESULT");
add(l1);
add(tf1);
 add(l2);
add(tf2);
add(l3);
add(tf3);
```

```
add(b1);
add(b2);
}
public void actionPerformed(ActionEvent ae)
{
 if(ae.getSource()==b1)
 {

try
{
int a=Integer.parseInt(tf1.getText());
int b=Integer.parseInt(tf2.getText());
int c=a/b;
tf3.setText(""+c);
}
catch(NumberFormatException ex)
{
tf3.setText(" ");
JFrame f=new JFrame();
JOptionPane.showMessageDialog(f,"Enter only numbers");
repaint();
}
catch(ArithmeticException ex)
{
tf3.setText(" ");
JFrame f=new JFrame();
JOptionPane.showMessageDialog(f,"Enter second value non zero");
```

```
repaint();
}
}
else
{
 tf1.setText("");
 tf2.setText("");
 tf3.setText("");
 msg="";
 repaint();
}
}
public void paint(Graphics g)
{
g.drawString(msg,30,70);
}
}
```

Set-10

---

a)//grid,layout

```
 import java.awt.*;
import java.io.*;
import java.util.*;
import javax.swing.*;
public class DisplayTable
{
public static void main(String[] args)
```

```
{
JFrame f = new JFrame("FILE READING EXAMPLE");
try
{
Scanner st=new Scanner(new FileReader("Emp.txt"));
 st.useDelimiter("\\s*,\\s*");
while(st.hasNext())
{
f.add(new Label(st.next()));
f.setLayout(new GridLayout(6,2));
f.setSize(400,200);
f.setVisible(true);
}
}
 catch (Exception ex)
{
System.out.println("Error reading file ");
}
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

b)

```
  //jlist
 import javax.swing.*;
 public class ListExmp
 {
```

```
ListExmp(){
JFrame f= new JFrame();
DefaultListModel<String> l1 = new DefaultListModel<>();
l1.addElement("Item1");
l1.addElement("Item2");
l1.addElement("Item3");
l1.addElement("Item4");
JList<String> list = new JList<>(l1);
list.setBounds(100,100, 75,75);
f.add(list);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public static void main(String args[])
{
new ListExmp();
}
}
//jtable
   import javax.swing.*;
public class TableEx
{
JFrame f;
TableEx()
```

```
{
f=new JFrame();
String data[][]={ {"501","ravi","50000"},
{"502","raju","70000"},
{"503","ramu","90000"}};
String column[]={"ID","NAME","SALARY"};
JTable jt=new JTable(data,column);
JScrollPane sp=new JScrollPane(jt);
f.add(sp);
f.setSize(300,400);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public static void main(String[] args)
{
new TableEx();
}
}
```