Unit I

1. Explain various loops used in JAVA and also write a program to calculate the following sum: 5+10+15+20+.....+100

```
public class DemoSum
{
public static void main(String args[])
{
int sum=0;
for(int i=5;i<=100;i+=5)
{
sum=sum+i;
}
System.out.println("sum="+sum);
}
}
```

2. Write a java program to print the square of first 10 even numbers.

```
public class EvenSquare
{
public static void main(String args[])
{
int sum=0;
for(int i=1;i<=20;i++)
{
if(i%2==0)
System.out.println("Square of "+i+"is
"+(i*i));
}
}
}
```

3. Write a program to find largest number amongst the three numbers using conditional or ternary operator.

Conditional Operator=>

```
public class MaxNumber
{
public static void main(String args[])
{
int a = 5, b = 10, c = 15;
System.out.println("Amongst
"+a+","+b+","+c);
if(a>=b && a>=c)
System.out.println(a+" is the largest
Number");

else if (b>=a && b>=c)
System.out.println(b+" is the largest
Number");
else

System.out.println(c+" is the largest
number");
}
}
```

Ternary Operator =>

```
public class MaxNumber
{
public static void main(String args[])
{
int n1 = 5, n2 = 10, n3 = 15, max;

max=(n1 > n2)?(n1 > n3 ? n1 : n3):(n2 > n3
? n2 : n3);

System.out.println("Largest number
among " + n1 +
                    ", " + n2 + " and " + n3 +
                      " is " + max + ". " );
}
}
```

4. Write an application program in java to Swap Two Numbers Using Third Variable.

```
 public class Swap
{
public static void main(String args[])
{
int a=5,b=6,temp;
System.out.println("Before Swap");
System.out.println("a= "+a+" b= "+b);
temp=a;
a=b;
b=temp;
System.out.println("After Swap");
System.out.println("a= "+a+" b= "+b);
}
}
```

5. Write a Java Program to Check Whether the Character is Vowel or Consonant using IF-ELSE.

```
 public class VowelConsonant
{
public static void main(String args[])
{
char ch = 'u';

    if(ch == 'a' || ch == 'e' || ch == 'i' || ch ==
'o' || ch == 'u' )
      System.out.println(ch + " is vowel");
    else
      System.out.println(ch + " is
consonant");
}
}
```

Unit II

1. What is an array? Write a program in java to sort elements of given array in ascending order.

```
 public class AscendingOrder
{
public static void main(String args[])
{
    int[] marks = {3, 5, 1, 2, 4};
    int temp, n;
    n = marks.length;
    System.out.print("The list of marks is:
");
    for(int i = 0; i< n; i++){
    System.out.print(marks[i]+ " ");        }
    for (int i = 0; i< n; i++){
    for (int j = i+1; j < n; j++){
    if (marks[i] > marks[j])   {
    temp = marks[i];
    marks[i] = marks[j];
    marks[j] = temp;          }}}
    System.out.print("\n List of marks
sorted in ascending order is: ");
    for (int i = 0; i< n; i++){
    System.out.print(marks[i]+" ");        }
}
}
```

2. Write a class student having data members name, roll_no and branch of student. Declare one constructor to initialize above data members and one display() method to display the information of a single student.

```
 class Student
{

 String name;
 int rollno;
 String branch;
 Student(String n,int r,String b)
```

```
{
 name=n;
 rollno=r;
 branch=b;
 }
 void display()
{
System.out.println("**Student
Information**");
System.out.println("Name: "+name);
System.out.println("Roll No: "+rollno);
System.out.println("Branch: "+branch);
}
}
class StudentInfo{
 public static void main(String args[])
{
 Student s1=new
Student("John",11,"CSE");
 s1.display();
 }
}
```

Unit III

1. Create a class named 'Member' having the following members:
   Data members
   1 - Name
   2 - Age
   3 - Salary
   It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, and salary to an employee and a manager by making an object of both of these classes and print all the respective data members of the classes.

```
class Member
{
 String name;
 int age;
 double salary;
 Member(String n,int a,double s)
{
  name=n;
  age=a;
  salary=s;
 }
 void printSalary()
{
System.out.println("Name: "+name);
System.out.println("Age: "+age);
System.out.println("Salary: "+salary);
}
}

class Employee extends Member
{
```

```
String specialization;
Employee(String name,int age,double
salary,String specialization)
{

super(name,age,salary);

this.specialization=specialization;

}

void displayEmployee()
{
System.out.println("\n*Employee
Information*");
printSalary();
System.out.println("Specialization:
"+specialization);
}
}
class Manager extends Member
{
String department;
Manager(String name,int age,double
salary,String department)
{
 super(name,age,salary);
 this.department=department;
}
void displayManager()
{
System.out.println("\n*Manager
Information*");
printSalary();
System.out.println("Department:
"+department);
}
}
class DemoInfo{
 public static void main(String args[])
{
 Employee e=new
Employee("Alexa",10,5600.00,"Google");
 e.displayEmployee();
 Manager m=new
Manager("Siri",15,59000.00,"Apple");
 m.displayManager();
 }
}
```

Unit IV

1. Write a program to handle array index out of bound exception.

```
public class ExcepArray
        {
                public static void
main(String args[])
                {
                try
                {
                int a[] = new int[2];

        System.out.println("Access
element three :" + a[3]);
                }
                catch
(ArrayIndexOutOfBoundsExceptio
n e)
                {

        System.out.println("Excepti
on thrown :" + e);
                }

        System.out.println("Out of
the block");
                }
        }
```

2. Write a java program to read character stream using file reader.

```
import java.io.*;
public class CharacterRead {

  public static void main(String args[])
throws IOException {
    FileReader in = null;

    try {
      in = new FileReader("input.txt");
      int c;
      while ((c = in.read()) != -1) {
            }
    }finally {
      if (in != null) {
        in.close();
      }
    }
  }
}
```

3. What is an Exception handling mechanism? Write a program to handle following exception

i) Divide by zero

```
public class ExpArith{
 public static void main(String
args[])
{
        try     {
        //code that may raise
exception
        int data=100/0;
        }
        catch(ArithmeticException
e)
                {

        System.out.println(e);
                }
  //rest code of the program
  System.out.println("rest of the
code...");
 }
}
```

ii) Array index out of bound

```
public class ExcepArray
        {
                public static void
main(String args[])
                {
                try
                {
```

```
                int a[] = new int[2];

            System.out.println("Access
        element three :" + a[3]);
                }
                catch
        (ArrayIndexOutOfBoundsExceptio
        n e)
                {

            System.out.println("Excepti
        on thrown :" + e);
                }

            System.out.println("Out of
        the block");
                }
            }
```

4. Write a program to create and delete a file object.

Create File=>

```
import java.io.*;
public class FileDemo {
  public static void main(String[] args) {

    try {
      File file = new File("javaFile123.txt");
      if (file.createNewFile()) {
        System.out.println("New File is
created!");
      } else {
        System.out.println("File already
exists.");
      }
    } catch (IOException e) {
      System.out.println(e);
    }

  }
}
```

Delete file=>

```
        import java.io.File; // Import the
File class
        public class DeleteFile
        {
        public static void main(String[]
args)
        {
        File myObj = new
File("filename.txt");
        if (myObj.delete())
        {
        System.out.println("Deleted the
file: " + myObj.getName());
        }
        else { System.out.println("Failed to
delete the file."); }
        }
        }
```

5. Write a java program to write the data into file using writer class.

OR

Write a program that creates a text file called test.txt and writes on it the string "some text written on a file".

```
 import java.io.*;
class WriteFileContent
{
public static void main(String args[])
{
try
{
FileWriter f=new FileWriter("test.txt");
f.write("some text written on a file");
f.close();
}
catch(Exception e)
{
System.out.println(e);
}}}
```

Unit V

1. Write a program to display string on appletviewer and rectangle using drawRect() method.

```
 import java.awt.*;
import java.applet.*;
/*
<applet code="RectangleN" width=300
Height=300>
</applet>
*/
public class RectangleN extends Applet
{
    public void paint(Graphics g)
    {
        g.drawRect(10,10,60,50);
        g.drawString("Object Oriented
Programming",10,80);
    }
}
```

2. Write an applet program to draw circle using drawOval() method and also display the string “SGBAU” inside the circle.

```
import java.awt.*;
import java.applet.*;
/*
<applet code="CircleN" width=300
Height=300>
</applet>
*/
public class CircleN extends Applet
{
    public void paint(Graphics g)
    {
        g.drawOval(100,100,100,100);
        g.drawString("SGBAU",130,160);
    }
}
```

3. Write a program to pass employee name and id number to an applet.

```
 /*<APPLET CODE = ParamPassing.class
WIDTH = 300 HEIGHT = 250>
        <param NAME = yourName VALUE
= John>
        <param NAME = yourID VALUE =
1035> </applet>*/
        import java.awt.*;
        import java.applet.*;
        public class ParamPassing
extends Applet {
        String name;
        int id;
        public void start() {
        String str;
        name                              =
getParameter("yourName");
        if (name == null)
        name = "not found";
        str = getParameter("yourID");
        try {
        if (str != null)
                id = Integer.parseInt(str);
        else id = 0;
        } catch (NumberFormatException
e) {}
        }
        public void paint(Graphics g) {

        g.drawString("*****Employee
Information*****", 10, 10);
                g.drawString("Name:
"+name, 10, 30);
                g.drawString("ID:  "+id,  10,
50);
        }}
```

Unit VI

1. Write a program to create checkbox

```
 /*<applet code="CheckboxDemo.class"
width=200 height=300></applet>*/
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class CheckboxDemo extends
Applet implements ItemListener
{ Checkbox c1=null; Checkbox c2=null;
Checkbox c3=null;
public void init()
{
c1=new Checkbox("JAVA"); c2=new
Checkbox("M3"); c3=new
Checkbox("DS");
add(c1); add(c2); add(c3);
c1.addItemListener(this);
c2.addItemListener(this);
c3.addItemListener(this);
}
public void paint(Graphics g){
g.drawString("JAVA:
"+c1.getState(),10,80);
g.drawString("M3:
"+c2.getState(),10,160);
g.drawString("DS:
"+c3.getState(),10,240); }
public void itemStateChanged(ItemEvent
ie)
{ repaint();
} }
```

2. Write a program to create a list on appletviewer.

```
import java.awt.*;
import java.applet.*;
/*<applet code="OperList.class"
width=300 height=300>
</applet>*/
public class OperList extends Applet
{
public void init()
{
Label l=new Label("City Name:");
List l1=new List(2);//rows
l1.add("Amravati");
l1.add("Akola");
l1.add("Nagpur");
l1.add("Mumbai");
l1.add("Pune");
add(l);
add(l1);
}
}
```

3. Write an applet program to draw two button objects with label as Red and Green, when user click on a particular button then background color will be change.

```
import java.awt.*;
import java.awt.event.*;
public class OperCard extends Frame
implements ActionListener
{
Panel cardPanel;
Panel p1,p2;
Panel buttonP;
Button b1,b2;
CardLayout cLayout;
public void opercard()
{
cardPanel=new Panel();
cLayout=new CardLayout();
cardPanel.setLayout(cLayout);
p1=new Panel();
p1.setBackground(Color.red);
p2=new Panel();
p2.setBackground(Color.green);

b1=new Button("Red");
b1.addActionListener(this);
b2=new Button("Green");
```

```
b2.addActionListener(this);
buttonP=new Panel();
buttonP.add(b1);
buttonP.add(b2);

cardPanel.add(p1,"b1");
cardPanel.add(p2,"b2");

setLayout(new BorderLayout());
add(buttonP,BorderLayout.SOUTH);
add(cardPanel,BorderLayout.CENTER);
setVisible(true);
setSize(500,400);
setTitle("CardDemo");
addWindowListener(new
WindowAdapter(){
public void windowClosing(WindowEvent
we){
System.exit(0);}});
}
public void actionPerformed(ActionEvent
ae)
{
if(ae.getSource()==b1)
  cLayout.show(cardPanel,"b1");
if(ae.getSource()==b2)
  cLayout.show(cardPanel,"b2");
}
public static void main(String args[])
{
OperCard obj=new OperCard();
obj.opercard();
}
}
```

4. Write a program to demonstrate the key event handlers.

```
 /*<applet code="KeyDemo.class"
width=300 height=300> </applet>*/
        import java.awt.*;
        import java.awt.event.*;
        import java.applet.*;
        public class KeyDemo extends
Applet {
        public void init(){
        addKeyListener(new KeyAdapter(){
        public void keyPressed(KeyEvent
ke){
        showStatus("Key Pressed");}
        public void keyReleased(KeyEvent
me){
        showStatus("Key Released");      }
        });              }}
```

5. Write a Java Application to create a file menu with menu items like New, Open, Save, Print and Close.

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

public class OperMenu1 extends Frame
implements ActionListener
{
public void operMenu()
{
setTitle("MenuBar");
setSize(250,150);
MenuBar menuBar=new MenuBar();
setMenuBar(menuBar);
Menu fileMenu=new Menu("File");

MenuItem newAction=new
MenuItem("New");
MenuItem openAction=new
MenuItem("Open");
MenuItem saveAction=new
MenuItem("Save");
MenuItem printAction=new
MenuItem("Print");
MenuItem exitAction=new
MenuItem("Exit");

newAction.addActionListener(this);
openAction.addActionListener(this);
saveAction.addActionListener(this);
```

```
printAction.addActionListener(this);
exitAction.addActionListener(this);

fileMenu.addSeparator();
fileMenu.add(newAction);
fileMenu.addSeparator();
fileMenu.add(openAction);
fileMenu.addSeparator();
fileMenu.add(saveAction);
fileMenu.addSeparator();
fileMenu.add(printAction);
fileMenu.addSeparator();
fileMenu.add(exitAction);

menuBar.add(fileMenu);
setVisible(true);

addWindowListener(new
WindowAdapter(){
public void windowClosing(WindowEvent
we){
System.exit(0);}});
}
public void actionPerformed(ActionEvent
ae)
{
String action=ae.getActionCommand();
if(action.equals("New"))
{
System.out.println("New");
}
else if(action.equals("Open"))
{
System.out.println("Open");
}
else if(action.equals("Exit"))
{
System.exit(0);
}
else if(action.equals("Save"))
{
System.out.println("Save");
}
else if(action.equals("Print"))
{
System.out.println("Print");
}
}
public static void main(String args[])
{
OperMenu1 obj=new OperMenu1();
obj.operMenu();
}
}
```

6. Explain Mouse Listener and mouse motion listener interface in event delegation model.

Ans=> Mouse Listener program

```
 /*<applet code="AdapterDemo.class"
width=300 height=300></applet>*/
        import java.awt.*;
        import java.awt.event.*;
        import java.applet.*;
        public class AdapterDemo
extends Applet {
        int xcord,ycord;
        public void init(){
        addMouseMotionListener(new
MouseDemo(this));
        }
        public void paint(Graphics g){
        g.drawString("("+xcord+","+ycord+
")",xcord,ycord);
        }}
Mouse motion listener
class MouseDemo extends
MouseMotionAdapter
{
        AdapterDemo d;
        MouseDemo(AdapterDemo d)
        {
                this.d = d;
        }
```

```
        public void
mouseMoved(MouseEvent me){
                d.xcord = me.getX();
                d.ycord = me.getY();
                d.repaint();
        }}
```