//OBJECT AND CLASS:

```
import java.util.Scanner;
class circle
{
int r;
float area;
float perimeter;
public void perimeter()
{
Scanner s1=new Scanner(System.in);
System.out.println("Enter the radius");
r=s1.nextInt();
area=3.14f*r*r;
System.out.println("Area is:"+area);
perimeter=2*3.14f*r;
System.out.println("Perimeter is:"+perimeter);
}
}
class vino
{
public static void main(String[] args)
{
circle c1= new circle();
c1.perimeter();
}
}
```

//PARAMETERIZED CONSTRUCTOR:

```
import java.io.*;
class student
{
int id;
```

```
String name;
student( int i , String S )
{
id=i;
name=S;
}
void display()
{
System.out.println("id="+id);
System.out.println("name="+name);
}
public static void main(String[] args)
{
student s1=new student(1001,"vinoliya");
student s2=new student(1002,"vinisha");
s1.display();
s2.display();
}
}
```

//DEFAULT CONSTRUCTOR:

```
import java.io.*;
class student
{
int id,age;
void display()
{
System.out.println("Id="+id);
System.out.println("Age="+age);}
public static void main(String[]args)
```

```
{
student s1=new student();
student s2=new student();
s1.display();
s2.display();
}
}
```

COPY CONSTRUCTOR:

```
import java.io.*;
class student
{
int id;
int age;
student(int i,int a)
{
id = i ;
age = a ;
}
student(student s)
{
id = s.id ;
age = s.age ;
}
void display()
{
System.out.println("id="+id);
System.out.println("age="+age);
}
public static void main(String args[])
{
student s1=new student(1001,17);
```

```
student s2=new student(s1);
s1.display();
s2.display();
}
}
```

//CONSTRUCTOR OVERLODING:

```
import java.io.*;
class student
{
int id;
int age;
String name;
student(int i , String s)
{
id=i;
name=s;
}
student(int i , int a , String s)
{
id=i;
name=s;
age=a;
}
public static void main(String[] args)
{
student s1= new student(1001,"liya");
student s2= new student(1002,18,"vinoliya");
System.out.println("id=" + s1.id );
System.out.println("name=" + s1.name );
System.out.println("id=" + s2.id );
System.out.println("name=" + s2.name );
```

```
System.out.println("age=" + s2.age );
}
}
```

//SINGLE INHERITANCE:

```
import java.io.*;
import java.util.*;
class parent
{
int no, age;
public void displaybase()
{
Scanner sc=new Scanner(System.in);
System.out.println("enter number and age:");
no=sc.nextInt();
age=sc.nextInt();
System.out.println("no="+no+"age="+age);
}
}
class child extends parent
{
String name;
public void displayderived()
{
Scanner sc=new Scanner(System.in);
System.out.println("enter name:");
name=sc.nextLine();
System.out.println("name="+name);
}
public static void main(String[] args)
{
```

```
child ch=new child();
ch.displayderived();
ch.displaybase();
}
}
```

//MULTIPLE INHERITANCE:

```
import java.io.*;
import java.util.*;
class parent
{
int no,age;
public void display()
{
Scanner sc=new Scanner(System.in);
System.out.println("read no and age");
no=sc.nextInt();
age=sc.nextInt();
System.out.println("no="+no);
System.out.println("age="+age);
}
}
class D1extendsparents
{
int m1,m2,tot,avg;
public void process1()
{
Scanner sc=new Scanner(System.in);
System.out.println("read m1&m2");
m1=sc.nextInt();
m2=sc.nextInt();
```

```
tot=m1+m2;
avg=tot/2;
System.out.println("m1="+m1);
System.out.println("m2="+m2);
System.out.println("total="+tot);
System.out.println("avg="+avg);
}
}
class D2 extends D1
{
String name;
public void display()
{
Scanner sc=new Scanner(System.in);
System.out.println("read name");
name=sc.nextLine();
System.out.println("name="+name);
}
public static void main(String[] args)
{
D2 ob=new D2();
ob.display();
ob.process1();
}
}
```

//HIERARCIAL INHERITANCE:

```
import java.io.*;
import java.util.*;
class parent
{
int no, age;
```

```
public void process()
{
Scanner sc=new Scanner(System.in);
System.out.println("read no & age");
no=sc.nextInt();
age=sc.nextInt();
System.out.println("no ="+no);
System.out.println("age ="+age);
}
}
class D1 extends parent
{
int m1,m2,m3,tot,avg;
public void process1()
{
Scanner sc=new Scanner(System.in);
System.out.println("read m1&m2");
m1=sc.nextInt();
m2=sc.nextInt();
tot=m1+m2;
avg=tot/2;
System.out.println("m1="+m1);
System.out.println("m2="+m2);
System.out.println("total="+tot);
System.out.println("average="+avg);
}
}

class D2 extends parent
{
String name;
```

```
public void display()
{
Scanner sc=new Scanner(System.in);
System.out.println("read name");
name=sc.nextLine();
System.out.println("name="+name);
}
}
class hier
{
public static void main(String args[])
{
D1 ob=new D1();
D2 ob1=new D2();
ob1.display();
ob.process();
ob.process1();
}
}
```

//ACCESS PROPERTY:

```
import java.io.*;
class base
{
String colour="White";
}
class derived extends base
{
String colour="Green";
public void display()
{
System.out.println("Base colour="+super.colour);
```

```
System.out.println("Deried colour="+colour);
}
}
class test1
{
public static void main(String[]args)
{
derived d=new derived();
d.display();
}
}
```

//ACCESS METHOD:

```
import java.io.*;
class animal
{
public void eat()
{
System.out.println("Eat non veg");
}
}
class dog extends animal
{
public void eat()
{
System.out.println("Eat eat.....");
}
public void bark()
{
System.out.println("Barking...");
}
public void display()
```

```
{
super.eat();
eat();
bark();
}
}
class test2
{
public static void main(String[]args)
{
dog d=new dog();
d.display();
}
}
```

//ACCESS CONSTRUCTOR:

```
import java.io.*;
class base
{
int no;
int age;
base (int no,int age)
{
this.no=no;
this.age=age;
}
}
class d1 extends base
{
String name;
d1(int no,int age,String name)
{
```

```
super(no,age);
this.name=name;
}
public void display()
{
System.out.println("No="+no);
System.out.println("Age="+age);
System.out.println("Name="+name);
}
}
class test3
{
public static void main(String[]args)
{
d1 d= new d1(01,18,"Princy");
d.display();
}
}
```

//ABSTRACT CLASS:

```
import java.io.*;
abstract class arithmetic
{
abstract void calculate(int a,int b);
}
class add extends arithmetic
{
void calculate(int a, int b)
{
int c=a+b;
System.out.println("Sum="+c);
}
```

```
}
class subtract extends arithmetic
{
void calculate(int a, int b)
{
int diff=a-b;
System.out.println("Difference="+diff);
}
}
class multiply extends arithmetic
{
void calculate(int a,int b)
{
int mul=a*b;
System.out.println("Product="+mul);
}
}
class arithmeticmain
{
public static void main(String[] args)
{
add a=new add();
subtract b= new subtract();
multiply c=new multiply();
a.calculate(80,96);
b.calculate(99,43);
c.calculate(23,17);
}
}
```

//MULTIPLE INHERITANCE THROUGH INTERFACE:

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

```
interface area
{
final float pi=3.14f;
float compute (float x,float y);
}
class rectangle implements area
{
public float compute (float x, float y)
{
return(x*y);
}
}
class circle implements area
{
public float compute(float x,float y)
{
return(pi*x*x);
}
}
class test
{
public static void main(String[]args)
{
rectangle rect= new rectangle();
circle cir=new circle();
area a;
a=rect;
System.out.println("Area of rectange:"+a.compute(10,20));
a=cir;
System.out.println("Area of circle:"+a.compute(10,0));
}
```

//ONE INHERITANCE THROUGH INTERFACE:

```
mport java.io.*;
interface a
{
void meth1();
void meth2();
}
interface b extends a
{
void meth3();
}
class myclass implements b
{
public void meth1()
{
System.out.println("Implement metn1()");
}
public void meth2()
{
System.out.println("Implement meth2()");
}
public void meth3()
{
System.out.println("Implements meth3()");
}
}
class inter
{
public static void main(String[] args)
{
myclass obj=new myclass();
```

```
obj.meth1();
obj.meth2();
obj.meth3();
}
}
```

//PACKAGE

Note pad-1

```
package calc;
public class A
{
public int add(int x,int y)
{
return x+y;
}
public int sub(int x,int y)
{
return x-y;
}public int mlt(int x,int y)
{
return x*y;
}
public int div(int x,int y)
{
return x/y;
}
public static void main(String[] args)
{
A s1=new A();
System.out.println(s1.sub(10,4));
}
}
```

```
import pack.*;
import java.util.*;
class B
{
public static void main(String args[])
{
int x,y;
A obj = new A();
Scanner m=new Scanner(System.in);
System.out.println("ent the digits");
x=m.nextInt();
y=m.nextInt();
System.out.println("sum is:" + obj.add(x,y));
System.out.println("subtracted val is:" + obj.sub(x,y));
}
}
```

//EXCEPTION HANDLING:

```
public class excpt
{
	public static void main(String args[]) throws ArithmeticException
	{
		try
		{
			int a=43/0; // error stmt since 43 cannot be divided by zero
		}
		catch(ArithmeticException e)
		{
			System.out.println(e);
		}
		finally
		{
```

```
                    System.out.println("end prgm");
                    }
          }
}
```

//USER DEFINE EXCEPTION:

```
class MyException extends Exception {
   public MyException(String s)
   {

      super(s);
   }
}
public class Main
{
   public static void main(String args[])
   {
      try
                    {

         throw new MyException("exception handled");
      }
      catch (MyException ex)
                    {
         System.out.println("Caught");



         System.out.println(ex.getMessage());
      }
   }
}
```

//THREAD EXTENDING BY THREAD CLASS:

```
import java.io.*;
class two extends Thread
{
public void run()
{
int i;
for (i=1;i<=5;i++)
{
System.out.println(i +" * 2 = "+(i*2));
}
}
}
class three extends Thread
{
public void run()
{
int i;
for(i=1;i<5;i++)
{
System.out.println(i+" * 3 = "+(i*3));
}
}
}
class wel extends Thread
{
public void run()
{
int i;for(i=1;i<5;i++)
{
System.out.println("welcome");
}
```

```
}
}
class threadexample
{
public static void main(String args[])
{
two t= new two();
three th=new three();
wel w=new wel();
t.start();
th.start();
w.start();
}
}
```

//ICLONABLE INTERFACE:

```
class StudentimplementsCloneable
{
int rno;
String name;
public Student(int rno, String name)
{
   this.rno = rno;
   this.name = name;
}
public Object clone() throws CloneNotSupportedException
{
   return super.clone();
}

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

```
Student s = new Student(95, "Kishore");
System.out.println("original object is:");
System.out.println(s.rno + " " + s.name);
try
{
  Student s1 = (Student)s.clone();
  System.out.println("cloned object is:");
  System.out.println(s1.rno + " " + s1.name);
}
catch (Exception e)
{
  System.out.println("excptn caugth");
}
}
}
```

//WARPPERS CLASS:

```
class Wrps
{
	public void primitivetoobj()
	{
		int a=50;
		Integer a1=a;    //Autoboximg
		System.out.println("integer obj is:" + a +" "+ a1);

		float b= 3.14f;
		Float b1=b;
		System.out.println("float obj is:" + b+ " "+b1);

		byte c=100;
		Byte c1=c;
		System.out.println("byte obj is:" + c+ " " +c1);
```

```
        char d= 'T';
        Character d1=d;
        System.out.println("charater obj is:" + d+ " "+d1);

        short e=4000;
        Short e1=e;
        System.out.println("short obj is:" + e+ " "+e1);

        long f=1200000000L;
        Long f1=f;
        System.out.println("long obj is:" + f+ " "+f1);

        double g=12.77d;
        Double g1=g;
        System.out.println("double obj is:" + g+ " "+g1);

        boolean h=true;
        Boolean h1=h;
        System.out.println("boolean obj is:" + h +" " +h1);

}
public void objtoprimitive()
{
        Integer a1=Integer.valueOf(60);
        int a2=a1;   //Unboxing
        System.out.println("value of int variable:" + a1+ " "+ a2);

        Float b1=Float.valueOf(3.14f);
        float b2=b1;
        System.out.println("value of float variable:" + b1+" "+b2);
```

```
		/*Byte c1=Byte.valueOf(100);
		byte c2=c1;
		System.out.println("byte obj to primitive type is:" + c1+" " +c2);*/

		Character d1=Character.valueOf('H');
		char d2=d1;
		System.out.println("value of character variable:" + d1+" "+d2);

		/*Short e1=new Short(300);
		short e2=e1;
		System.out.println("short obj to primitive type is:" + e1+" "+e2);*/

		Long f1=Long.valueOf(1200000000L);
		long f2=f1;
		System.out.println("value of long variable:" + f1+ " "+f2);

		Double g1=Double.valueOf(12.77);
		double g2=g1;
		System.out.println("value of double variable:" + g1+ " "+g2);

		Boolean h1=Boolean.valueOf(true);
		boolean h2=h1;
		System.out.println("value of boolean variable:" + h1+" " +h2);
	}


}
class Wrp
{
	public static void main(String[] args)
```

```
	{
		Wrps t=new Wrps();
		t.primitivetoobj();
		t.objtoprimitive();
	}
}
```

//INPUT STREAM AND OUTPUT STREAM:

```
 import java.io.*;
class Opt
{
public static void main(String[] args)
{
try
{
String M="welcome";
FileOutputStream f1=new FileOutputStream("D://text.txt");
byte[]b1= M.getBytes();
f1.write(b1);
f1.close();
System.out.println("written successfully");
}
catch(Exception e)
{
System.out.println("caunt write");
}
Inp j=new Inp();
j.input();
}
}
class Inp
{
```

```
public void input()
{
try
{
FileInputStream f1=new FileInputStream("D://text.txt");
int i=0;
while((i=f1.read())!=-1)
{
System.out.print((char)i);
}
f1.close();
}
catch(Exception e)
{
System.out.println("excptn caught");
}
}
}
```

//READER AND WRITTER:

```
import java.io.*;
class Wrt
{
public static void main(String[] args)
{
try
{
FileWriter r=new FileWriter("D://info.txt");
String Z="Sistians";
r.write(Z);
r.close();
System.out.println("writting done");
```

```
}
catch(Exception e)
{
System.out.println("caunt writer");
}
Rdr g=new Rdr();
g.rerd();
}
}
class Rdr
{
public void rerd()
{
try
{
FileReader r=new FileReader("D://info.txt");
int i=0;
System.out.print("string inside info file is:");
while((i=r.read())!=-1)
{
System.out.print((char)i);
}
}
catch(Exception e)
{
System.out.println("caunt read");
}
}
}
```

//DATA INPUT AND OUTPUT STREAM

```
import java.io.*;
class Dos
{
public static void main(String[] args)
{
try
{
FileOutputStream d1=new FileOutputStream("D://alan.txt");
DataOutputStream dos1= new DataOutputStream(d1);
dos1.writeInt(67);
System.out.println("written successfully");
dos1.close();
d1.close();
}
catch(Exception e)
{
System.out.println("caunt write");
}
Dis k=new Dis();
k.shar();
}
}
class Dis
{
public void shar()
{
try
{
FileInputStream d1=new FileInputStream("D://alan.txt");
DataInputStream dis1= new DataInputStream(d1);
int y=dis1.readInt();
```

```
System.out.println("the read message is:"+y);
dis1.close();
d1.close();
}
catch(Exception e)
{
System.out.println("caunt read");
}
}
}
```

//LIFE CYCLE OF APPLET

```
import java.applet.Applet;
import java.awt.*;
/*<applet code ="Lifcyc" width=400 height=400>
</applet> */
public class Lifcyc extends Applet
{
public void init()
{
System.out.println("init called");
}
public void start()
{
System.out.println("start called");
}
public void paint()
{
System.out.println("pain called");
}
public void stop()
{
```

```
System.out.println("stop called");
}
public void destroy()
{
System.out.println("destroy called");
}
}
```

//APPLET PARAMETER:

```
import java.applet.Applet;
import java.awt.Graphics;
/*<applet code ="Studdet" width=300 height=300>
<param name="Name" value="princy">
<param name="Dept" value="IT">
<param name="RegNo" value=41120142>
</applet>*/
public class Studdet extends Applet
{
String s1,s2,s3;
public void init()
{
s1=getParameter("Name");
s2=getParameter("Dept");
s3=getParameter("RegNo");
}
public void paint(Graphics g)
{
g.drawString("Reading parameter frm applet",20,20);
g.drawString("Name:"+s1,20,40);
g.drawString("Dept:"+s2,20,60);
g.drawString("RegNo:"+s3,20,80);
}
```

```
}
```

EVENT HANDLING:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code ="EventApplet" width=400 height=400>
</applet> */
public class EventApplet extends Applet implements ActionListener
{
Button b;
TextField tf;
public void init()
{
tf=new TextField();
tf.setBounds(40,60,150,20);
b=new Button("Click");
b.setBounds(80,180,60,60);
add(b);
add(tf);
b.addActionListener(this);
setLayout(null);
}
public void actionPerformed(ActionEvent e)
{
tf.setText("Hey there....");
}
}
```

MOUSE AND KEY EVENTS IN JAVA

```
/* program to handle Mouse and Key events */
```

```
import java.awt.*;
import java.awt.event.*;
public class MouseListenerExample extends Frame implements MouseListener{
   Label l;
   MouseListenerExample(){
      addMouseListener(this);

      l=new Label();
      l.setBounds(20,50,100,20);
      add(l);
      setSize(300,300);
      setLayout(null);
      setVisible(true);
   }
   public void mouseClicked(MouseEvent e) {
      l.setText("Mouse Clicked");
   }
   public void mouseEntered(MouseEvent e) {
      l.setText("Mouse Entered");
   }
   public void mouseExited(MouseEvent e) {
      l.setText("Mouse Exited");
   }
   public void mousePressed(MouseEvent e) {
      l.setText("Mouse Pressed");
   }
   public void mouseReleased(MouseEvent e) {
      l.setText("Mouse Released");
   }
```

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

SWING CONTROL:

```
import javax.swing.*;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="Swing.class" width=300 height=300>
</applet>*/
public class Swing extends JApplet implements ActionListener
{
JButton b1;
JTextField t1;
JLabel l1;
public void init()
{
b1=new JButton("Click");
b1.setBounds(80,160,60,20);
t1=new JTextField();
t1.setBounds(40,60,150,20);
l1=new JLabel("Sathyabama");
l1.setBounds(20,40,100,20);
add(l1);
add(t1);
add(b1);
b1.addActionListener(this);
setLayout(null);
}
public void actionPerformed(ActionEvent e)
```

```
{
t1.setText("HELLO");
}}
```