1. Classes and Object

//class and objects

import java.io.\*;

import java.util.\*;

class circle {

float r, area, perimeter;

public void read() {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the radius:");

r = sc.nextInt();

}

public void calc() {

area = r \* r \* (float)Math.PI;

perimeter = 2 \* r \* (float)Math.PI;

}

public void display() {

System.out.println("Area =" + area);

System.out.println("Perimeter =" + perimeter);

}

}

class area

{

public static void main(String[] args)

{

circle ob= new circle();

ob.read();

ob.calc();

ob.display();

}

}

2.a) No argument constructor

//no\_argument constructor

class no\_arg

{

int n;

String s;

no\_arg()

{

n=7;

s="Arun";

}

public static void main (String args[])

{

no\_arg ob = new no\_arg();

System.out.println("No ="+ob.n);

System.out.println("Name ="+ob.s);

}

}

2.b) Parameterized constructor

//parameterized constructor

class sample

{

int n;

String s;

sample(int id,String name)

{

n=id;

s=name;

}

}

class Parameterized

{

public static void main(String args[])

{

sample ob = new sample(7,"Arun");

System.out.println("No ="+ob.n);

System.out.println("Name ="+ob.s);

}

}

2.c) Copy constructor

// copy constructor

import java.io.\*;

import java.util.\*;

class student

{

int id;

String name;

student(int i, String s)

{

id = i;

name=s;

}

student(student s)

{

id=s.id;

name=s.name;

}

void display()

{

System.out.println("Id ="+id);

System.out.println("Name ="+name);

}

public static void main(String args[])

{

student s1 = new student(7,"Arun");

student s2 = new student(s1);

s1.display();

s2.display();

}

}

2.d) Constructor overloading

//constructor overloading

import java.io.\*;

import java.util.\*;

class overloading

{

int id,age;

String name;

overloading(int i, String s)

{

id=i;

name=s;

}

overloading(int i,int a, String s)

{

id = i;

age = a;

name = s;

}

void display()

{

System.out.println("Reg No="+id);

System.out.println("Name ="+name);

System.out.println("Age ="+age);

}

public static void main(String args[])

{

overloading s1 = new overloading(10,18,"Arun");

overloading s2 = new overloading(7, 18,"Ash");

s1.display();

s2.display() }

3.a) Single inheritance

//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 no and age :");

no = sc.nextInt();

age = sc.nextInt();

System.out.println("No=" +no+"Age="+age);

}

}

public class single extends parent

{

String name;

public void display()

{

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[])

{

single ch = new single();

ch.display();

ch.displaybase(); }}

3.b) Multilevel inheritance

//multilevel inheritance

import java.io.\*;

import java.util.\*;

class parent1

{

String name;

int regno;

public void displaybase()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter your Name and Regno :");

name = sc.nextLine();

regno = sc.nextInt();

System.out.println("Name ="+name);

System.out.println("Regno ="+regno);

}

}

class child1 extends parent1

{

int age;

public void display()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter your age:");

age = sc.nextInt();

System.out.println("Age="+age);

}

}

class child2 extends child1

{

String location;

public void display1()

{

Scanner sc= new Scanner(System.in);

System.out.println("Enter your loction");

location = sc.nextLine();

System.out.println("Location="+location);

}

}

class multilevel

{

public static void main(String args[])

{

child2 ch = new child2();

ch.displaybase();

ch.display();

ch.display1();

}

}

3.c) Hierarchical inheritance

//hierarchical inheirtance

import java.io.\*;

import java.util.\*;

class baseA

{

int no,age;

public void input()

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter no and age:");

no = sc.nextInt();

age = sc.nextInt();

}

}

class derivedB extends baseA

{

String name;

public void input1()

{

name = "Arun";

System.out.println("Name="+name);

}

}

class derivedC extends baseA

{

public void display()

{

System.out.println("Reg no="+no);

System.out.println("Age="+age);

}

}

class hierarchical

{

public static void main (String[] args)

{

derivedB ob = new derivedB();

derivedC ob1 = new derivedC();

ob.input1();

ob1.input();

ob1.display();

}

}

4.a) Super Keyword – Parent variable

//super keyword

import java.io.\*;

class animal

{

String color="white";

}

class dog extends animal

{

String color="grey";

public void display()

{

System.out.println(color);

System.out.println(super.color);

}

}

class parentvariable

{

public static void main(String args[])

{

dog d = new dog();

d.display();

}

}

4.b) Super Keyword – Parent Method

//super keyword

import java.io.\*;

class animal1

{

public void eat1()

{

System.out.println("Eating Non Veg...");

}

}

class dog extends animal1

{

@java.lang.Override

public void eat1() {

System.out.println("Eating Meat");

}

public void bark()

{

System.out.println("barking");

}

public void work()

{

super.eat1();

eat1();

bark();

}

}

class parentmethod

{

public static void main(String args[])

{

dog d = new dog();

d.work();

}

}

4.c) Super Keyword - Parent Constructor

//super keyword

import java.io.\*;

class base {

base()

{

System.out.println("base created");

}

}

class d1 extends base

{

d1()

{

super();

System.out.println("derived created");

}

}

class parentconstructor

{

public static void main(String args[])

{

d1 d = new d1();

}

}

5. Final Keyword

class final1

{

int a=10;

int b=7;

final int c = a+b;

public void run()

{

System.out.println("Sum="+c);

}

public static void main(String args[])

{

final1 ob = new final1();

ob.run();

}

}

6. Abstract Class

//abstraction

import java.util.\*;

abstract class bank

{

public abstract void Intrest();

}

class sbi extends bank

{

public void Intrest()

{

int intrest,P,R,N;

Scanner s1= new Scanner(System.in);

System.out.println("ent the values");

P=s1.nextInt();

R=s1.nextInt();

N=s1.nextInt();

intrest=(P\*R\*N)/100;

System.out.println("sbibank\_intrest is:" + intrest);

}

}

class indianbank extends bank

{

public void Intrest()

{

int intrest,P,R,N;

Scanner s1= new Scanner(System.in);

System.out.println("ent the values");

P=s1.nextInt();

R=s1.nextInt();

N=s1.nextInt();

intrest=(P\*R\*N)/100;

System.out.println("indianbank\_intrest is:" + intrest);

}

}

class abst

{

public static void main(String[]args)

{

bank s2;

s2= new sbi();

s2.Intrest();

s2= new indianbank();

s2.Intrest();

}

}

7. Area using interface or multiple inheritance

import java.lang.\*;

interface area

{

static 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);

}

}

public class multiple

{

public static void main(String args[])

{

rectangle rect = new rectangle();

circle c = new circle();

area a;

a = rect;

System.out.println("Area of rectangle="+a.compute(10,20));

a = c;

System.out.println("Area of circle="+a.compute(3,0));

}

}

8. Package

package arith;

public class Arith

{

public int add(int a, int b)

{

return a+b;

}

public int sub(int a, int b)

{

return a-b;

}

public int mul(int a, int b)

{

return a\*b;

}

public int div(int a, int b)

{

return a/b;

}

}

import arith.\*;

public class sample {

public static void main(String[] args)

{

Arith ss = new Arith();

System.out.println("Sub:"+ss.sub(10,3));

}

}

9. Exception Handling

//exception handling

class exp

{

public static void main(String args[])

{

try

{

String a = null;

System.out.println("The length of the string is:"+a.length());

}

catch(NullPointerException e)

{

System.out.println("Exception caught");

System.out.println("Exception:\n"+e);

}

}

}

10. User Defined exception

import java.util.\*;

class UserDefinedException extends Exception

{

public UserDefinedException (String s)

{

super(s);

}

}

class Uexp

{

static void validate (int age) throws UserDefinedException

{

if(age<18)

{

throw new UserDefinedException("Sorry");

}

}

public static void main(String[] args)

{

try {

Scanner sc = new Scanner(System.in);

System.out.println("Enter your age:");

int

a1 = sc.nextInt();

validate(a1);

System.out.println("You are eligible");

}

catch(UserDefinedException e)

{

System.out.println("Exception caught");

System.out.println(e.getMessage());

}

}

}

11. Thread using runnable interface

import java.io.\*;

class two implements Runnable

{

public void run()

{

int i;

for(i=1;i<=5;i++)

{

System.out.println(i+"\*2="+(i\*2));

}

}

}

class three implements Runnable

{

public void run()

{

int i;

for(i=1;i<=5;i++)

{

System.out.println(i+"\*3="+(i\*3));

}

}

}

class runnableexample

{

public static void main(String[] args)

{

two tw = new two();

Thread t1 = new Thread(tw);

t1.start();

three th = new three();

Thread t2 = new Thread(th);

t2.start();

}

}

12. Cloneable interface

class Student implements Cloneable

{

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(28, "Arun");

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");

}

}

}

13. Wrapper 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();

}

}

14. a) File Input & Output Stream

//file input

import java.io.\*;

import java.io.FileInputStream;

import java.io.InputStream;

public class ipstream

{

public static void main(String[] args)

{

byte[] ary = new byte[100];

try

{

InputStream ip = new FileInputStream("sample.txt");

System.out.println("Available size = "+ip.available());

ip.read(ary);

String data = new String(ary);

System.out.println("File content:"+data);

ip.close();

}

catch (Exception e)

{

System.out.println("Error = "+e);

}

}

}

//output stream

import java.io.\*;

public class opstream {

public static void main(String[] args) {

String data = "Hello Arun";

try {

OutputStream op = new FileOutputStream("hello.txt");

byte[] ary = data.getBytes();

op.write(ary);

System.out.println("File written successfully");

op.close();

} catch (Exception e) {

System.out.println("Error=" + e);

}

}

}

14.b) Reader and Writer

import java.io.\*;

public class readereg

{

public static void main(String args[])

{

try{

Reader re = new FileReader("out2.txt");

int data;

data = re.read();

while(data != -1)

{

System.out.println((char)data);

data = re.read();

}

re.close();

}

catch (Exception e)

{

System.out.println("Error="+e.getMessage());

}

}

}

//writer class

import java.io.\*;

class writereg

{

public static void main(String args[]) throws Exception

{

try

{

FileWriter wt = new FileWriter("out2.txt");

String data = "Hello world";

wt.write(data);

System.out.println("Written successfully");

wt.close();

}

catch (Exception e)

{

System.out.println(e);

}

}

}

14. c) Data input and output stream

import java.io.\*;

public class dipstream

{

public static void main(String args[]) throws IOException

{

InputStream ip = new FileInputStream("out3.txt");

DataInputStream dis = new DataInputStream(ip);

int count = ip.available();

System.out.println("Size="+ip.available());

byte[] ary = new byte[count];

dis.read(ary);

for(byte bt:ary)

{

char ch = (char) bt;

System.out.println(ch+"");

}

dis.close();

ip.close();

}

}

import java.io.\*;

public class dopstream

{

public static void main(String args[]) throws IOException

{

OutputStream op = new FileOutputStream("out3.txt");

DataOutputStream dos = new DataOutputStream(op);

String s = "Hello Arun";

byte[] b = s.getBytes();

dos.write(b);

dos.flush();

dos.close();

op.close();

System.out.println("Data Written successfully");

}

}

15. Lifecycle of applet

import java.applet.\*;

import java.awt.\*;

/\*<applet code="graphicseg.class" height=200 width=200>

</applet>\*/

public class graphicseg extends Applet

{

public void init()

{

setBackground(Color.green);

System.out.println("Init stage is invoked");

}

public void start()

{

System.out.println("Start stage is invoked");

}

public void paint(Graphics g)

{

g.setColor(Color.white);

g.drawRect(25, 35, 50, 60);

g.fillRect(25, 35, 50, 60);

}

public void stop()

{

System.out.println("Stop is invoked");

}

public void destroy()

{

System.out.println("Applet is destroyed");

}}

16. Passing parameters to applet

import java.applet.\*;

import java.awt.\*;

/\*<applet code ="passvalue.class" width=300 height=300>

<param name="sub1" value="java"/>

<param name="sub2" value="ads"/>

</applet>\*/

public class passvalue extends Applet

{

public void paint(Graphics g)

{

String s1 =getParameter("sub1");

String s2 =getParameter("sub2");

g.drawString(s1, 50,50);

g.drawString(s2, 50,80);

}

17. AWT Controls

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*<applet code ="eventapplet.class" 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);

tf.setText("Heyy theree..");

b=new Button("Click");

b.setBounds(80,180,60,60);

add(b);

add(tf);

b.addActionListener(this);

setLayout(null);

}

public void actionPerformed(ActionEvent e)

{

}

}

18. Mouse and Keyboard

mouse

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*<applet code ="Mse" width=500 height=500>

</applet> \*/

public class Mse extends Applet implements MouseListener,MouseMotionListener

{

String msg="";

int mx=100,my=100;

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);

}

public void mousePressed(MouseEvent e)

{

msg="down";

mx=e.getX();

my=e.getY();

repaint();

}

public void mouseReleased(MouseEvent e)

{

msg="Up";

mx=e.getX();

my=e.getY();

repaint();

}

public void mouseClicked(MouseEvent e)

{

msg="Mouseclicked";

mx=e.getX();

my=e.getY();

repaint();

}

public void mouseEntered(MouseEvent e)

{

msg="mouseEntered";

repaint();

}

public void mouseExited(MouseEvent e)

{

msg="mouseExited";

repaint();

}

public void mouseMoved(MouseEvent e)

{

mx=e.getX();

my=e.getY();

showStatus("mousemoved at"+mx+" "+my);

}

public void mouseDragged(MouseEvent e)

{

mx=e.getX();

my=e.getY();

showStatus("mouseDragged at"+mx+" "+my);

}

public void paint(Graphics g)

{

g.drawString(msg,100,100);

}

}

keyboard

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\* <applet code="key.class" width=500 height=500>

</applet> \*/

public class key extends Applet implements KeyListener {

String msg="";

int kx = 100, ky = 100;

public void init()

{

addKeyListener(this);

}

public void keyPressed(KeyEvent ke)

{

showStatus(" Key Down");

}

public void keyReleased(KeyEvent ke)

{

showStatus(" Key Up");

}

public void keyTyped(KeyEvent ke) {

msg += ke.getKeyChar();

repaint();

}

public void paint(Graphics g)

{

g.drawString(msg, kx, ky);

}}

19. Swing controls

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(80,160,160,20);

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");

}

}