Shubham Tomar

BTECH CSE 6TH SEM

1702910157 KIET

1.

Class box{

private Double height;

private Double width;

private Double depth;

Public box(){

,This.depth=depth;

This.width=width;

This.height=height;

}

Public double getvolume(){

Return depth\*height\*volume;

}

}

//driver code

Public class Handon assignment{

Public static void main(String args[]){

Box box=new Box(16,9,7);

System.out.println(Box.getvolume());

}

}

2.

Class calculator{

Public static int powerint (int num1,int num2){

Return (int)Math.pow(num1,num2);}

Public static double powerdouble(double num1,double num2){

Return (double)Math.pow(num1,num2)}

}

//driver code

Public class handon assignment{

Public static void main(String args[]){  
 System.out.println(calculator.powerint(12,6));

System.out.println(calculator.powerdouble(63,12));

}}

3.

Class Author{

Private String name;

Private String email;

Private Char gender;

Public Author(String name, String email, char gender){

Super();

This.name=name;

This.email=email;

This.gender=gender;

}

Public String getName() {

Return name;}

Public String getEmail() {  
 return email;}

Public Char getGender(){

Return gender;}

@override

Public String toString(){

Return “Author[name=”+name+”,,email=”+email+”,gender=”+gender+”]”;

}}

Class Book{

Private String name;

Private Author author;

Private double price;

Private int qtyinstock;

Public Book(String name, Author author, double price, int qtyinstock){

Super();  
This.name=name;

This.author=author;

This.price=price;

This.qtyinstock=qtyinstock;}

Public double getPrice (){

Return price;}

Public void setPrice(double price){

This.price=price;}

Public int getQtyinstock(){

Return qtyinstock;}

Public void getQtyinstock(int qtyinstock){

This.qtyinstock=qtyinstock;}

Public String getName(){

Return name;}

Public Author getAuthor(){

Return author;}

@override

Public String toString(){

Return “Book[name=”+name+”,author=”+author+”,price=,+price+”,qtyinstock=”+qtyinstock+”]”;}}

4.

Class Animal{

Public void eat(){ System.out.println(“Animal eat”);}

Public void sleep(){ System.out.println(“Animal sleep”);}}

Class Bird extends Animal(){

@override

Public void eat(){ System.out.println(“Bird eat”);}

@override

Public void sleep(){ System.out.println(“Bird sleep”);}

Public void fly(){ System.out.println(“Bird fly”);}

}

//driver code

Public class Handon assignments{

Public static void main(string[] args){

Animal animal=new Animal();

animal.eat();

animal.sleep();}

Bird bird=new Bird();

bird.eat();

bird.sleep();

bird.fly();

}}

5.

Public Class person{

Protected String name;

Public String setName(String name){

This.name=name;}

Public String getName(){

Return name;}

@override

Public string tostring(){

Return “person[name=”+name+”]”;}}

Public Class employee extends person{

Private double annual salary;

Private int yearofjoining;

Private string nationalinsuranceno;

Public employee(double annualsalary, int yearofjoining ,String nationalinsuranceno){

Super(name);

this.annualsalary=annualsalary;

this.yearofjoining=yearofjoining;

this.nationalinsuranceno=nationalinsuranceno;

}

Public double getAnnualsalary(){ return annualsalary;}

Public int getYearofjoining(){ return yearofjoining;}

Public String getNationalinsuranceno(){ return nationalinsuranceno;}

@override

Public String getName() { return name;}

@override

Public String toString() {

Return “Employee[annualsalary=”+annualsalry+”,yearofjoining=”+yearofjoining+”,nationalinsuranceno=”+nationalinsuranceno+”,name=”+name+”]”;

}}

6.

Public Class Fruit{

Protected String name;

Protected String taste;

Protected Int size;

Public void eat(){

Public fruit(){ this.name=name;}

Public String getName(){ return name;}

Public fruit(){ this.taste=taste;}

Public String getTaste(){ return taste;}

}}

Public Class Apple extends fruit{

@oveeride

Public void eat(){};

}

Public class Orange extends fruit{

@override

Public void eat(){};

}

//driver code

Public class Handon assignment{

Public static void main(String[] args){

New fruit.eat();

New Apple.eat();

New orange.eat();}}

7.

Public class shape{

Public void draw(){};

Public void erase(){};}

Public class Circle extends shape {

@override

Public void erase(){ System.out.println(“Erasing Circle”);}

@override

Public void draw(){System.out.println(“Drawing Circle”); }}

Public class Square extends shape{

@override

Public void draw(){ System.out.println(“Drwaing Square”);}

@override

Public void erase() { System.out.println(“Erasing Square”);}}

Public class Triangle extends shape{

@override

Public void draw(){System.out.println(“Drawing Triangle”);}

@override

Public void erase(){System.out.println(“ Erasing Triangle ”);}}

//driver code

Public class Handon assignment{

Public static void main(String[] args){

Shape c=new Circle();

Shape t=new Triangle();

Shape s=new Square();

c.draw();

c.erase();

t.draw();

t.erase();

s.draw();

s.erase();

}}

8.concatenate two strings java program

Public class stringconcatenate{

Public static void main(String[] args){

String s1;

String s2;

S1=s1.toLowerCase();

S2=s2.toLowerCase();

String s;

S=s1+s2;

//s=s1.concat(s2);

System.out.println(s);}}

9.

Public class Handon assignment{

Public static boolean ispalindrome(String input1){

Int i=0, j=input1.length()-1;

While(i<j){

If(input1.charAt(i)!=input1.charAt(j))

Return false;

i++;

j--;

Return true;}

//Driver code

Public static void main(String[] args){

System.out.println(ispalindrome(“MADAM”));}}

10.

Public class Handon assignment{

Public static String newstring(String input1){

String output=” ”;

If(input1.length()<=1){ output=input1;}

Else{ for(int i=0;i<input1.length();i++){

Output=output+input1.charAt(0)+input1.charAt(1);}}}

//@driver code

Public static void main(String[] args){

System.out.println(newstring(“wipro”));}}

public class Handon assignment{

Public static string firsthalf(String input1){

Int n=input1.length();

If(n%2==0){ for(int i=0;i<n/2;i++){ return i;}}

Else{ return NULL;}

}

//driver code

Public static void main(string[] args){

System.out.println(firsthalf(“Tomcat”));}}

12.

Public class Handon assignment{

Public static void main(string[] args){

String input1=”wipro”;

Int n=input1.length();

Input1=input1.substring(1,n-1);

System.out.println(input1);}}

13.

Public class Handon assignment{

Public static string shortlongshort(String input1,String input2){

Int n=input1.length();

Int m=input2.length();

If(n<m){ return input1+input2+input1;}

Else{ return input2+input1+input2;}}

//driver code

Public static void main(String[] args){

System.out.println(shortlongshort(“hi,hello”));}}

14.

Public class Handon assignment{

Public static string lastcharx(String input1){

Int n=input1.length();

If(input1.charAt(0)==x)

Input1=input1.substring(1,input1.length());

If(input1.charAt(n-1)==x)

Input1=input1.substring(0,n-1);}

//@driver code

Public static void main(string[] args){

System.out.println(lastcharx(“xhix”));}}

15.

Public class Handon assignment{

Public static void main(String[] args){

String input1=”ab\*cd”;

String[] strs=input1.split(“.[\\\*]+.”);

StringBuffer sb=new StringBuffer();

For(String x: strs)

Sb.append(x);

System.out.println(sb);}}

16.

Public class Handon assignment{

public static string 1to12to2 (String input1,String input2){

Int n=input1.length();

Int m=input2.length();

StringBuffer sb=new StringBuffer();

For(int i=0;i<n;i++)

Sb.append(input1.charAt(i)).append(input2.charAt(i));}

//@driver code

Public static void main(String[] args)

System.out.println((“wipro”,”hello”));}}

17.

Public abstract class Compartment{

Public abstract void notice();}

Public class FirstClass extends Compartment{

@override

Public void notice(){

System.out.println(“Notice: you are in the first class”);}}

Public class Ladies extends Compartment{

@override

Public void notice(){

System.out.println(“you are in the ladies class”);}}

Public class General extends Compartment{

@override

Public void notice(){

System.out.println(“Notice:you are in the general class”);}}

Public class Luggage extends Compartment{

@override

Public void notice(){

System.out.println(“Notice: you are in the luggage in the class”);}}

Import java.util.Random.\*;

Public class TestCompartment{

Public static void main(String[] args){

Compartment[] compartments=new Compartment[10];

Random rand=new Random();

For(int i=0;i<10;i++){

Int randomNum=rand.nextInt((4-1)+1)+1;

If(randomNum==1) compartments[i]=new FirstClass();

Else if(randomNum==2) compartment[i]=new Ladies();

Else if(randomNum==3)compartments[i]=new General();

Else if(randomNum==4) compartment[i]=new Luggage();

Compartments[i].notice();

}}}

18.

Package music.\*;

Public interface playable{

Void play();}

Package music.string.\*;import music.playable.\*;

Public class Veena implements playable{

//override

Public void play(){ System.out.println(“playing veena”);}}

Package music.wind.\*; import music.playable.\*;

Public class Saxophone implements playable{

@override

Public void play(){ System.out.println(“playing Saxophone”);}}

Class Test{

Public static void main(String[] args){

Playable Veena=new veena(); veena.play();

Playable Saxophone = new saxophone(); saxophone.play();

}}}

1. public class Handon assignment{

Public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.println(“enter the no of element in an array”);

Int N=sc.nextInt;

Int[] arr =new Int[N]

System.out.println(“enter the element in the array”);

Try{

For(int i=0;i<N;i++)

arr[i]=sc.nextInt();

System.out.println(“enter the index of the array element you want to access”);

Int index=sc.nextInt();

System.out.println(“the array element at index”+index+”=”+arr[index]);

System.out.println(“the array element accessed successfully”);

}

Catch(arrayindexoutofbound e){

System.out.println(“java.lang.arrayinexoutofboundsexception”);}

Catch(NumberFormatException e){

System.out.println(“java.lang.NumberFormatException”);}

Sc.close();}}

20.

Public class Handon asssignment{

Public static void main(String[] args){  
int n=args.length();

Int[] arr=new int[n];

Int sum=0;

Double avg=0;

Try{

For(int i=0;i<n;i++) arr[i]=Integer.parseInt(args[i]);

For(int i=0;i<n;i++)

Sum+=arr[i];

Avg=sum/n;

}

Catch(ArithematicException e){ System.out.println(“ArithematicException”);}

Catch(NumberFormatException e){ System.out.println(”NumberFormatException”);}

Catch(InputMisMatchException e){ System.out.println(“InputMisMatchException”);}

System.out.println(“Sum:”+sum);

System.out.println(“Average:”+avg);}}

21

Public class Handon assignment{

Public static void main(String[] args){

Scanner sc=new Scanner(System.in);

For(int i=0;i<2;i++){

String name=””;

Int subA,=0,subB=0,subC=0;

Try{

name=sc.nextLine();

If(sc.hasNextInt()) subA=sc.nextInt(); else throw new NumberFormatException();

If(sc.hasNextInt()) subB=sc.nextInt(); else throw new NumberFormatException();

If(sc.hasNextInt()) subC=sc.nextInt(); else throw new NumberFormatException();

If(subA<0) throw new NegativeValuesException();

If(subA>100) throw new ValuesOutOfRangeException();

If(subB<0)throw newNegativeValueException();

If(subB>100)throw new ValuesOutOfRangeException();

If(subC<0)throw new NegativeValueException();

If(subc>100)throw new ValuesOutOfRangeException();}

Catch(ArithematicException e){ System.out.println(“e.getmessage());}

Catch(NegativeValueException e){ System.out.println(“e.getmessage());}

Catch(ValuesOutOfRangeException e){ System.out.println(“e.getmessage”);}

System.out.println(“Name:”+name);

System.out.println(“Marks of Subject A:”+subA);

System.out.println(“Marks of Subject B:”+subB);

System.out.println(“Marks of Subject C:”+subC);

}

Sc.close();}}

22.

Public class Handon assignment{

Public static void main(String[] args){

String name=args(0);

Int age=Integer.parseInt(args(1));

If(age<18 ||age>=60)throw new InvalidAgeException();

System.out.println(“Name:”+name+”Age:”+age);

}}

Public class InvalidAgeException extends Handon assignment{

Public void InvalidAgeException(){ super();

System.outprintln(“Invalid Age”);}}

23.

Public class Handon assignment{

Public static void main(String[] args){

System.out.println(“Integer range:”);

System.out.println(“min:”+Integer.MIN\_VALUE);

System.out.println(“max:”+Integer.MAX\_VALUE);

System.out.println(“Double range:”);

System.out.println(“min:”+Double.MIN\_VALUE);

System.out.println(“max:”+Double.MAX\_VALUE);

System.out.println(“Long range:”);

System.out.println(“min:”+Long.MIN\_VALUE);

System.out.println(“max:”+Long.MAX\_VALUE);

System.out.println(“Short range:”);

System.out.println(“min:”+Short.MIN\_VALUE);

System.out.println(“max:”+Short.MAX\_VALUE);

System.out.println(“Byte range:”);

System.out.println(“min:”+Byte.MIN\_VALUE);

System.out.println(“max:”+Byte.MAX\_VALUE);

System.out.println(“Float range:”);

System.out.println(“min:”+Float.MIN\_VALUE);

System.out.println(“max:”+Float.MAX\_VALUE);

}}

24.

Public class Handon assignment{

Public static void main(String[] args){

Int num=Intger.parseInt(args[0]);

System.out.println(“Given Number:”+num);

System.out.println(“Binary equivalent:”+Integer.toBinaryString(num));

System.out.println(“Octal equivalent:”+Integer.toOctalString(num));

System.out.println(“Hexadecimal equivalent:”+Integer.toHexString(num));

}}

25.

Public class Handon assignment{

Public static void main(String[] args){

Scanner sc=new Scanner(System.in);

Int num=sc.nexInt();

String result=string.format(“%8s”,IntegertoBinaryString(num)).replace(‘ ’,’0’);

System.out.println(result);} sc.close();}}

26.

Public class Employee implements Cloneable{

Private String name;

Public Employee(String name){ this.name=name;}

Pubilc void setName(String name){ this.name=name;}

Public String getName(){ Return name;}

@override

Public Employee clone(){

Try{ return(Employee) super.clone();}

Catch(CloneNotSupportedException e){ System.out.println(“Clonning Not Allowed”); return this;}}}

Public class Handon assignment{

Public static void main(String[] args){

Employee emp=new Employee(“Shubham Tomar”);

Employee empClone=emp.clone();

empClone.setName(“prince”);

Syste.out.println(empClone.getName());

System.out.println(emp.getName());}}

1. IO STREAM /OBJECT SERIALIZATION

To copy contents from one file to another file and check the output

import java.io.\*;

Import java.io.Exception.\*;

Import java.io.File.\*;

Import java.io.FileReader.\*;

Import java.io.FileWriter.\*;

Import java.util.Scanner.\*;

Import java.io.BufferedReader.\*;

Import java..io.BufferedWriter.\*;

Public class Handon assignment{

Public static void main(String[] args) throws IOException{

Scanner sc=new scanner(System.in);

System.out.println(“Enter the input file name”);

String inputfilename=sc.nextLine();

System.out.println(“Enter the output file name”);

String outfilename=sc.nextLine();

File filein=new File(inputfilename);

File fileout=new File(outputfilename);

BufferedReader br=new BufferedReader(new FileReader(filein));

BufferedWriter bw=new BufferedWriter(new FileWriter(fileout));

Int ch;

While((ch=br.read())!=-1){

bw.write(ch);};

br.close();

bw.close();

sc.close();}}

28.

Import java.util.Sacnner.\*;

Import java.IOException.\*;

Import java.io.BufferedReader.\*;

Import java.io.FileReader.\*;

Import java.io.\*;

Import java.io.File.\*;

Public class Handon assignment{

Public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.println(“Enter the input file name”);

String inputfilename=sc.nextLine();

System.out.println(“Enter the character to be counted”);

Char ch=sc.nextLine().charAt(0);

File file=new File(inputfilename);

Int charcount=0;

BufferedReader br =new BufferedReader(new FileReader(file));

Int a;

Do{

a=br.read();

If(a>=65 && a<=90)a+=32;

If(ch>=32&&ch<=90) ch+=32;

If(a==ch) charcount++;}

While(a!=-1);

System.out.println(“file”+filename+”’has”+charcount+”instances of letter’”+ch+”’.”);

Br.close();

Sc.close();}}

29.

**Import java.io.\*;**

**Import java.io.BufferedReader.\*;**

**Import java.io.BuffeRedwriter.\*;**

**Import java.io.FileReader.\*;**

**Import java.io.FileWriter.\*;**

**Import java.ioException.\*;**

**Import java.util.Iterator.\*;**

**Import java.util.Map.\*;**

**Import java.util.TreeMap.\*;**

**Import java.util.Set.\*;**

**Import java.util.MapEntry.\*;**

**Import java.util.Scanner.\*;**

**Public class Handon assignment{**

**Public static void main(String[] args){**

**Scanner sc=new Scanner(System.in);**

**System.out.println(“Enter the input file name”);**

**String inputfilename=sc.nextLine();**

**System.out.println(“Enter the output file name”);**

**String outputfilename=sc.nextLine();**

**File filein=new File(inputfilename);**

**File fileout=new File(outputfilename);**

**BufferedReader br=new BufferedReader(new FileReader(filein));**

**BufferedWriter bw=new BufferedWriter(new FileWriter(fileout));**

**Map<String,Integer> map=new TreeMap<>();**

**String str;**

**While((str=br.readLine())!=null){**

**str=str.Trim();**

**String[] words=str.split();**

**For(String word:words){ if(!map.containsKey(word)) map.put(word,1); else map.put(word,map.get(word)+1);**

**}};**

**Set<Entry<String,Integer>>set=map.entrySet();**

**Iterator<Entry<String.Integer>>it=set.iterator();**

**While(it.hasNext()){**

**Entry<String,Integer>me=it.next();**

**bw.write(me.getKey()+”:”+me.getValue()+”\n”);**

**} br.close(); bw.close(); sc.close():}}**

Import java.io.FileInputsream.\*;

Import java.io.FileOutputStream.\*;

Import java.ioExceptiom.\*;

Import java.ObjectOutputStream.\*;

Import java.ObjectInputStream.\*;

Import java.util.Scanner.\*;

Public class Handon assignment{

Public static void main)String[] args){

Scanner sc=new Scanner(System.in);

Employee emp=new Employee();

System.out.println(“Enter name:”);

emp.setName(sc.nextLine());

System.out.println(“Enter DO.B:”);

Emp.setDob(sc.nextLine());

System.out.println(“Enter department:”);

Emp.setDepartment(sc.nextLine());

System.out.println(“Enter designation:”);

Emp.setDesgination(sc.nextLine());

System.out.println(“Enter salary:”);

Emp.setSalary(sc.nextDouble());

Sc.nextLine();

FileOutputStream fos=new FileOutputStream(“OutObject.txt”);

ObjectOutputStream oos=new ObjectOutputStream(fos);

oos.writeObject(emp);

FileInputStream fis=newFileInputStream(“OutObject.txt”);

ObjectInputStream ois=new ObjectInputStream(fis);

Employee emp2=(Employee) ois.readObject();

System.out.println(“Name:”+emp2.getName());

System.out.println(“D.O.B:”+emp2.getDob());

System.out.println(“Department:”+emp2.getDeaprtment());

System.out.println(“Designation:”+emp2.getDesignation());

System.out.println(“Salary:”+emp2.getSalary());

fos.close();

fis.close();

oos.close();

ois.close();

sc.close();}}

31.

Import java.util.List.\*;

Import java.util.Iterator.\*;

Import java.util.ArrayList.\*;

Public class Handon assignment{

Public static void main(String[] args){

List<String>list=new ArrayList<>();

list.add(“shubham”);

list.add(“tomar”);

list.add(“tomarshabh”);

Printall(list);}

Public static void Printall(List<String>list){

Iterator<String>it=list.iterator();

While(it.hasNext())

System.out.println(it.next());}}

32.

Import java.util.List.\*;

Import java.util.ArrayList.\*;

Import java.util.iterator.\*;

Public class Handon assignment{

Public static void main(String[] args){

List<Integer>list=new Array List<>();

list.add(“Integer”);

list.add(“float”);

list.add(“double”);

33.

Import java.util.ArrayList.\*;

Import java.util.List.\*;

Class MyArrayList<E>extends ArrayList<E>{

@Override

Public boolean add(E e){

If( e instanceof Integer|| e instanceof Float ||e instanceof Double){

Super.add(e); return true;}

Else{ throw new ClassCastException(“only integer, Float Double are allowed”);}}}

Public Handon assignment{

Public static vid main(String[] args){

List<object>list=new MyArrayList<>();

Try{ list.add(15); list.add(1.2F);list.add(3.145D);list.add(“test”);}

Catch(Exception e){e.printStackTrace();}

System.out.println(list);}}

34.

Import java.util.HashSet.\*;

Import java.util.Iterator.\*;

Public class Handon assignment{

Public static void main(String[] args){

HashSet<String> name=new HashSet<>();

set.add(“shubham”);

set.add(“tomar”);

set.add(“prince”);

set.add(“badshah”);

Iterator<String> it=set.Iterator();

While(it.hasNext())

System.out.println(it.next());}}

35.

Import java.util.Iterator.\*;

Import java.util.HashSet.\*;

Public class Country{

HashSet<String>H1=new HashSet<>();

Public HashSet<String> saveCountryNames(String CountryName){

H1.add(CountryName);

Return H1;}

Public String getcountry(String CountryName){

Iterator<String> it=H1.iterator();

While(it.hasNext()){

If(it.next().equals(CountyName()) return CountryName;} return NULL;}}

//test code(driver)

Public class Handon Assignment{

Public static void main(String[] args){

Country countries=new Country();

countries.saveCountryName(“japan”);

countries.saveCountryName(“india”);

System.out.println(“india:”+countries.getCountry(“india”));

System.out.println(“pakistan:”+countries.getCountry(“pakistan”));

}}

36.

Import java.util.Collection.\*;

Import java.util.TreeSet.\*;

Import java.util.Set.\*;

Import java.util.Iterator.\*;

Public class Handon assignment{

Public static void main(String[] args){

TreeSet<String>set=new TreeSet<>();

//Collection<String>set=new TreeSet<>(Collections.reverseorder());

set.add(“tomar”);

set.add(“shubham”);

set.add(“prince”);

Iterator<String>it=set.iterator();

String Query=”shubham”;

Boolean result=false;

While(it.hasNext())

If(it.next().equals(query)){

Result=true;

Break;}}

If(result) System.out.println(query+”exists”);

Else System.out.println(query+” do not exists”);}}

37.

Import java.util.TreeSet.\*;

Import java.util.Set.\*;

Import java.util.Iterator.\*;

Public class Country{

TreeSet<String> H1=new TreeSet<>();

Public TreeSet<String> saveCountryNames(String CountryName){

H1.add(CountyName); return H1;}

Public String getCountry(String CountryName){

Iterator<String>it=H1.iterator();

While(it.hasNext()){

If(it.next().equals(CountryName))

Return CountryName;} return NULL;}}

//Driver code

Public class Handon assignment{

Public static void main(String[] args){

Country countries=new Country();

countries.saveCountryNames(“japan”);

countries.saveCountryNames(“india”);

System.out.println(“india:”+countries.getCountry(“india”));

System.out.println(“pakistan:”+countries.getCountry(“pakistan”));

}}

**MULTITHREADING**

1.

Public class Handon assignment{

Public static void main(String[] args){

Thread t1=new Thread(“Scooby”){

Public void run(){

System.out.println(“im”+Thread.currentThread().getName());}};

Thread t2=new Thread(“Shaggy”){

Public void run(){

System.out.println(“im”+Thread.currentThread().getName());}};

t1.start();

t2.start();

}}

2.

Import java.util.Random.\*;

Public class Handon assignment implements Runnable {

Public static void main(String[] args){

Handon handon=new Handon();

Thread t1=new Thread(handon);

T1.start();

}

@override

Public void run(){

Randon random=new Random();

String[] colours={“white”,”blue”,”black”,”green”,”red”,”yellow”};

Int index;

While((index=random.nextInt(6))!=4){

System.out.println(colours[index]);}}}

3.

Public class Handon assignment Implements Runnable{

Public static void main(String[] args){

Handon handon =new Handon();

Static Thread t1=new Thread(handon());

t1.start();}

@override

Public void run(){

For(int i=0;i<=10;i++){

If(i==6)

Try{

t1.sleep(5000);}

Catch(InterruptedException e){

e.printStackTrace();}

System.out.println(i);}}}

4.

Public class Handon assignment Implements Runnable{

Public static void main(String[] args){

Handon handon=new Handon();

Static Thread t1=new Thread(handon,”t1”);

Static Thread t2=new Thread(handon,”t2”);

t1.start();

t2.start();}

@override

Public void run(){

Try{

If(Thread.currentThread().getName().equals(“t1”))

t2.join();}

Catch(InterruptedException e){

e.printStackTrace();}

For(int i=1;i<20;i++){

If(Thread.cuurentThread().getName().equals(“t1”)){

If(i%2==0) System.out.println(i);}

If(Thread.currentThread().getName().equals(“t2”)){

If(i%2==1) System.out.println(i);}

}}}

5.

Public class Handon assignment Implements Runnable{

Public static void main(String[] args){

Handon handon=new Handon();

Static Thread t1=new Thread(handon);

Static Thread t2=new Thread(handon);

Static Thread t3=new Thread(handon);

t1.setPriority(Thread.MAX\_PRIORITY);

t2.setPriority(Thread.MIN\_PRIORITY);

t3.setPriority(Thread.NORM\_PRIORITY);

t1.start();

t2.start();

t3.start();}

@override

Public void run(){

For(int i=0;i<100;i++)

System.out.println(Thread.currentThread().getName()+”:”+i);}}