NATIONAL INSTITUTE OF TECHNOLOGY

WARANGAL – 506 004

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

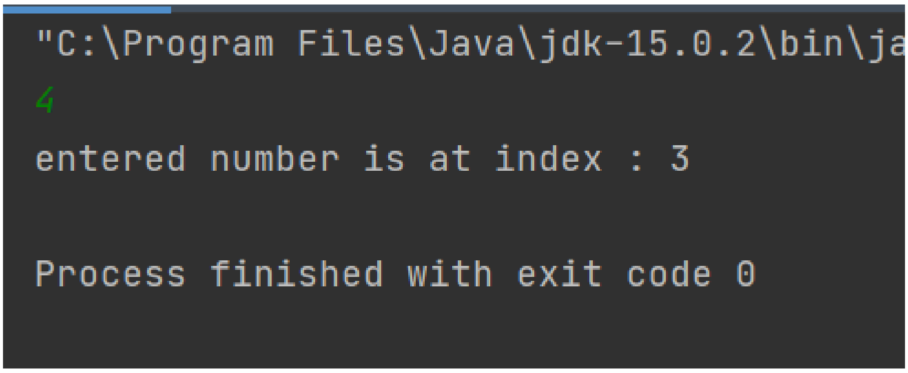
II B.Tech., II Semester

OOPS Lab Assignment

ASSIGNMENT 3

1. Write a Java Program to Implement Binary Search Algorithm.

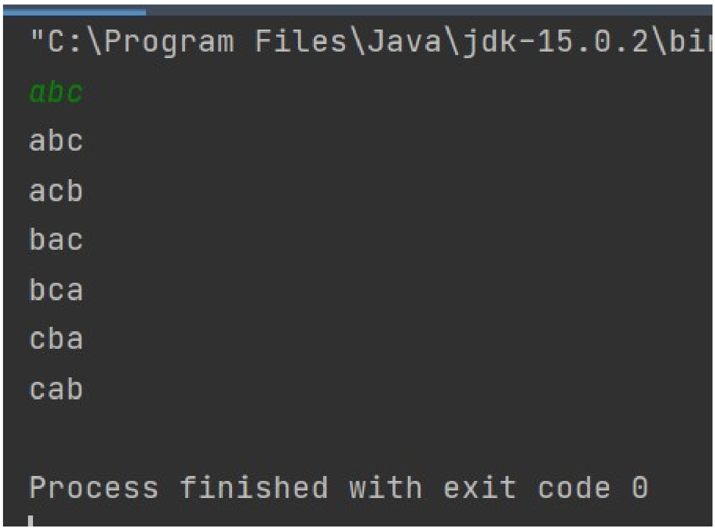
```
package com.deepak;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        int n,a[]={1,2,3,4,5};
        n=sc.nextInt();
        int l=0,r=a.length-1;
        while (l<r)
        {
            int m=(l+r)/2;
            if (a[m]<n)
                l=m+1;
            else
                r=m;
        }
        if (a[l]==n)
        System.out.println("entered number is at index : "+l);
        else
            System.out.println("number not found");
    }
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\ja
4
entered number is at index : 3

Process finished with exit code 0
```

2. Write a Java Program to Compute all the permutations of the string.

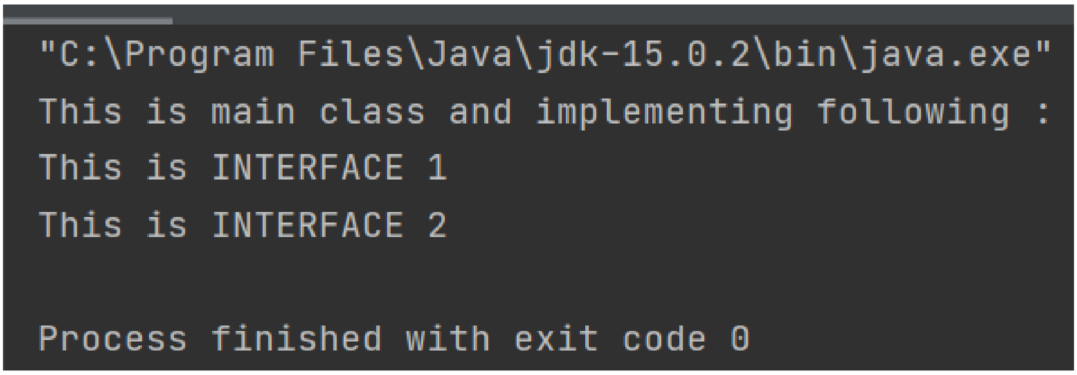
```
package com.deepak;
import java.util.Scanner;
import java.lang.String;


public class Main {

    public static String swap(String a, int i, int j)
    {
        char s[]=a.toCharArray();
        char t;
        t=s[i];
        s[i]=s[j];
        s[j]=t;
        return String.valueOf(s);
    }
    public static void permu(String s,int start, int end)
    {
        if (start==end-1) {
            System.out.println(s);
            return;
        }
        for (int i=start;i<end;i++)
        {
            s=swap(s,start,i);
            permu(s,start+1,end);
            s=swap(s,start,i);
        }


    }

    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        String a;
        a=sc.next();
        permu(a,0,a.length());

    }
}
```



```
"C:\Program Files\Java\jdk-15.0.2\bi
abc
abc
acb
bac
bca
cba
cab


Process finished with exit code 0
```

3. Write a Java Program to Implement multiple inheritance.

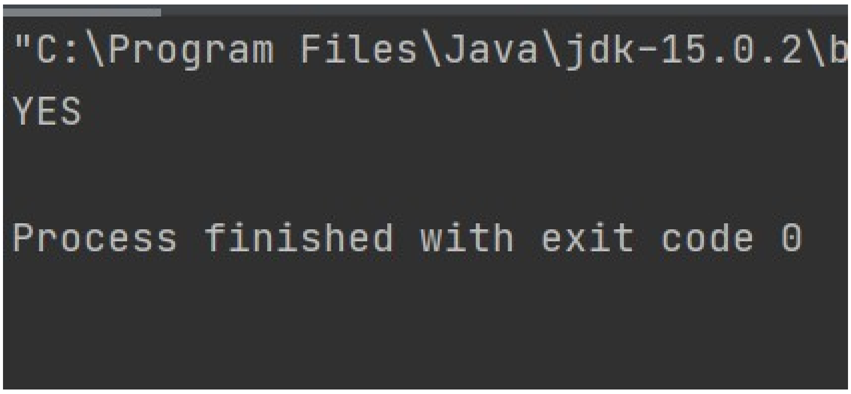
```
package com.deepak;
import java.util.*;
public class Main{

    public interface one{
        public static void display(){
            System.out.println("This is INTERFACE 1");
        }
    }

    public interface two{
        public static void display(){
            System.out.println("This is INTERFACE 2");
        }
    }

    public static class myclass implements one,two{

        public static void display(){
            System.out.println("This is main class and implementing following :");
            one.display();
            two.display();
        }
    }

    public static void main(String args[]){
        myclass myobj = new myclass();
        myobj.display();
    }

}
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe"
This is main class and implementing following :
This is INTERFACE 1
This is INTERFACE 2

Process finished with exit code 0
```

4. Write a Java Program to Detect loop in a LinkedList.

```
package com.deepak;
import java.util.Scanner;
import java.lang.String;


public class Main {
    static class node
    {
        public int data;
        public node next;
        node(int a,node ne)
        {
            data=a;
            next=ne;
        }
    }

    static void insert(node head, int t)
    {
        if (head==null)
        {
            head=new node(t,null);
            return;
        }
        node x=head;
        while (x.next!=null)
            x=x.next;
        x.next=new node(t,null);
    }

    static void print(node head)
    {
        if (head==null) {
            System.out.println();
            return;
        }
        System.out.print(head.data+" ");
        print(head.next);
    }
    static boolean has_loop(node head)
    {
        node fast=head,slow=head;
        while (fast!=null&&fast.next!=null)
        {
            fast=fast.next.next;
            slow=slow.next;
            if (fast==slow)
            {
                return true;
            }
        }
        return false;
    }
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        node head=null;
        head=new node(0,null);
        head.next=head;
        if (has_loop(head))
            System.out.println("YES");
        else
            System.out.println("NO");
    }
}
```

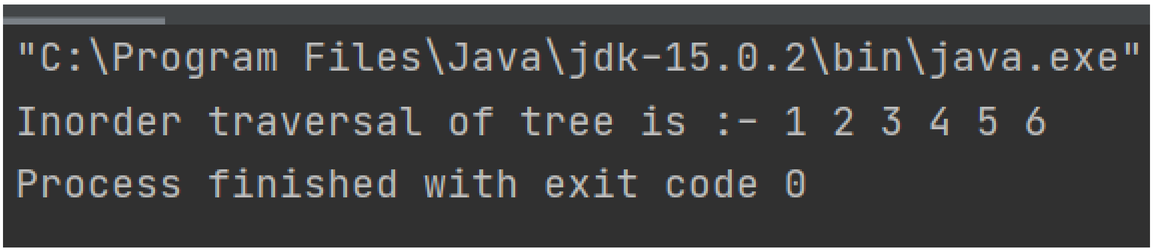


```
"C:\Program Files\Java\jdk-15.0.2\b
YES

Process finished with exit code 0
```

5. Write a Java Program to Implement Binary Tree Data Structure.

```java
package com.deepak;
import java.util.Scanner;
import java.lang.String;

class tnode
{
   int data;
   tnode left, right;
   tnode(int x, tnode l, tnode r)
   {
      data=x;
      left=l;
      right=r;
   }
}

public class Main {

   static tnode insert(tnode root,int data)
   {
      if (root==null)
      {
         return new tnode(data,null,null);
      }
      if (root.data>data)
      {
         root.left=insert(root.left,data);
      }
      else
         root.right=insert(root.right,data);
      return root;
   }
   static void inorder(tnode root)
   {
      if (root==null)
         return;
      inorder(root.left);
      System.out.print(root.data+" ");
      inorder(root.right);
   }
   public static void main(String[] args) {
      Scanner sc= new Scanner(System.in);
      tnode root=null;
      root=insert(root,4);
      root=insert(root,6);
```

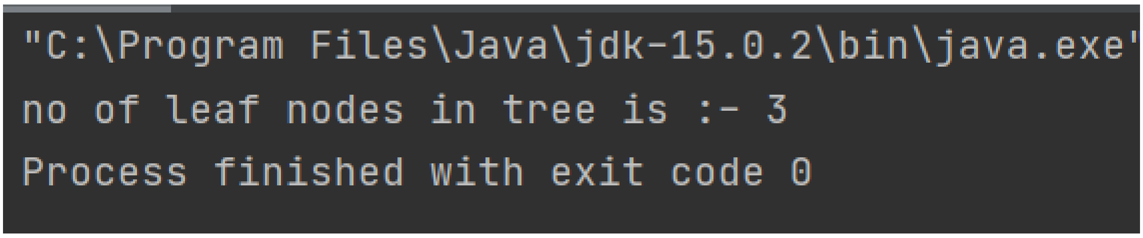
```
        root=insert(root,2);
        root=insert(root,3);
        root=insert(root,1);
        root=insert(root,5);
        System.out.print("Inorder traversal of tree is :- ");
        inorder(root);
    }
}
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe"
Inorder traversal of tree is :- 1 2 3 4 5 6
Process finished with exit code 0
```

6. Write a Java Program to Count number of leaf nodes in a tree.

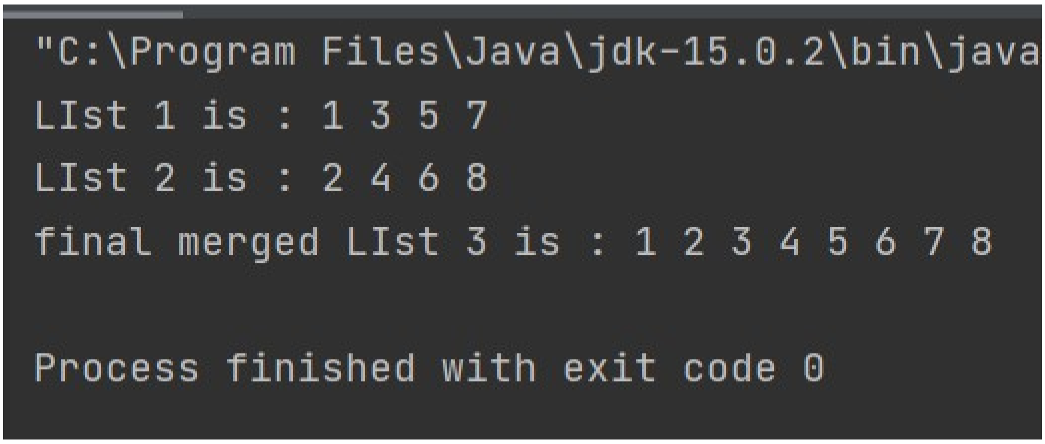
```
package com.deepak;
import java.util.Scanner;
import java.lang.String;

class tnode
{
    int data;
    tnode left, right;
    tnode(int x, tnode l, tnode r)
    {
        data=x;
        left=l;
        right=r;
    }
}

public class Main {

    static tnode insert(tnode root,int data)
    {
        if (root==null)
        {
            return new tnode(data,null,null);
        }
        if (root.data>data)
        {
            root.left=insert(root.left,data);
        }
        else
            root.right=insert(root.right,data);
        return root;
    }
    static int count_leaves(tnode root)
    {
        if (root==null)
            return 0;
        if (root.left==null&&root.right==null)
            return 1;
        return count_leaves(root.left)+count_leaves(root.right);

    }
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        tnode root=null;
        root=insert(root,4);
        root=insert(root,6);
        root=insert(root,2);
        root=insert(root,3);
        root=insert(root,1);
        root=insert(root,5);
        System.out.print("no of leaf nodes in tree is :- "+ count_leaves(root));

    }
}
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe'
no of leaf nodes in tree is :- 3
Process finished with exit code 0
```

7. Write a Java Program to Merge two lists.

```
package com.deepak;
import java.util.Scanner;
import java.lang.String;


public class Main {
    static class node
    {
        public int data;
        public node next;

        node(int a,node ne)
        {
            data=a;
            next=ne;
        }
    }

    static node insert(node head, int t)
    {
        if (head==null)
        {
            head=new node(t,null);
            return head;
        }
        node x=head;
        while (x.next!=null)
            x=x.next;
        x.next=new node(t,null);
        return head;
    }

    static void print(node head)
    {
        if (head==null) {
            System.out.println();
            return;
        }
        System.out.print(head.data+" ");
        print(head.next);
    }

    static node merge(node l1, node l2)
    {
        node h1=l1,h2=l2,l3=null;
        while (h1!=null && h2!=null)
        {
            if (h1.data<h2.data)
            {
                l3=insert(l3,h1.data);
                h1=h1.next;
            }
            else
            {
                l3=insert(l3,h2.data);
                h2=h2.next;
            }
        }
        while (h1!=null)
        {
            l3=insert(l3,h1.data);
            h1=h1.next;
        }
        while (h2!=null)
        {
            l3=insert(l3,h2.data);
            h2=h2.next;
        }
        return l3;
    }

    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        node list1=null,list2=null,list3=null;
        list1=insert(list1,1);
        list1=insert(list1,3);
        list1=insert(list1,5);
        list1=insert(list1,7);
        list2=insert(list2,2);
        list2=insert(list2,4);
        list2=insert(list2,6);
        list2=insert(list2,8);
        System.out.print("LIst 1 is : ");
        print(list1);
        System.out.print("LIst 2 is : ");
        print(list2);
        list3=merge(list1,list2);
        System.out.print("final merged LIst 3 is : ");
        print(list3);
    }
}
```

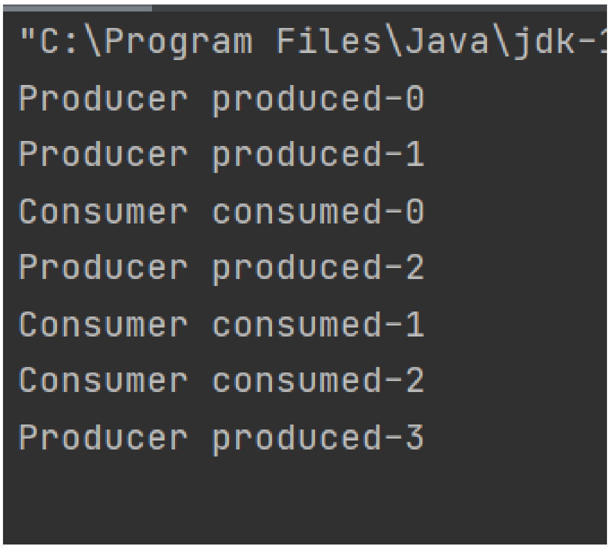


```
"C:\Program Files\Java\jdk-15.0.2\bin\java
LIst 1 is : 1 3 5 7
LIst 2 is : 2 4 6 8
final merged LIst 3 is : 1 2 3 4 5 6 7 8

Process finished with exit code 0
```

8. Write a java program that implements correctly the producer consumer problem using multithreading.

```
package com.deepak;
import java.util.*;

import java.util.LinkedList;

public class Main {
  public static void main(String[] args)
      throws InterruptedException
  {
    // Object of a class that has both produce()
    // and consume() methods
    final PC pc = new PC();

    // Create producer thread
    Thread t1 = new Thread(new Runnable() {
      @Override
      public void run()
      {
        try {
          pc.produce();
        }
        catch (InterruptedException e) {
          e.printStackTrace();
        }
      }
    });

    // Create consumer thread
    Thread t2 = new Thread(new Runnable() {
      @Override
      public void run()
      {
        try {
          pc.consume();
        }
        catch (InterruptedException e) {
          e.printStackTrace();
        }
      }
    });

    // Start both threads
    t1.start();
```

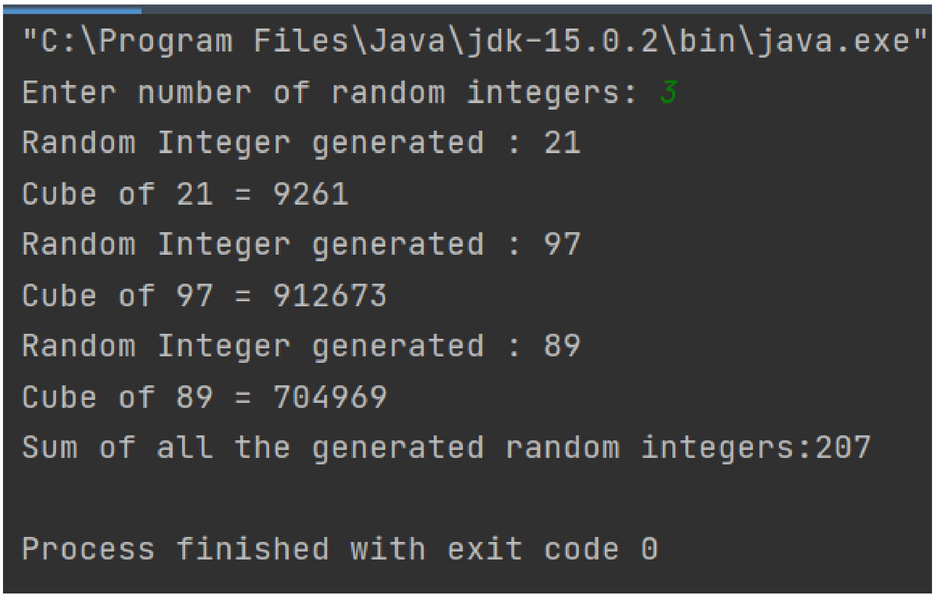
```
        t2.start();

        // t1 finishes before t2
        t1.join();
        t2.join();
    }

    // This class has a list, producer (adds items to list
    // and consumber (removes items).
    public static class PC {

        // Create a list shared by producer and consumer
        // Size of list is 2.
        LinkedList<Integer> list = new LinkedList<>();
        int capacity = 2;

        // Function called by producer thread
        public void produce() throws InterruptedException
        {
            int value = 0;
            while (true) {
                synchronized (this)
                {
                    // producer thread waits while list
                    // is full
                    while (list.size() == capacity)
                        wait();

                    System.out.println("Producer produced-"
                                + value);

                    // to insert the jobs in the list
                    list.add(value++);

                    // notifies the consumer thread that
                    // now it can start consuming
                    notify();

                    // makes the working of program easier
                    // to  understand
                    Thread.sleep(1000);
                }
            }
        }

        // Function called by consumer thread
        public void consume() throws InterruptedException
        {
            while (true) {
                synchronized (this)
                {
                    // consumer thread waits while list
                    // is empty
                    while (list.size() == 0)
                        wait();

                    // to retrive the ifrst job in the list
                    int val = list.removeFirst();

                    System.out.println("Consumer consumed-"
                                + val);
                    // Wake up producer thread
                    notify();

                    // and sleep
                    Thread.sleep(1000);
                }
            }
        }
    }
}
```



```
"C:\Program Files\Java\jdk-1
Producer produced-0
Producer produced-1
Consumer consumed-0
Producer produced-2
Consumer consumed-1
Consumer consumed-2
Producer produced-3
```

9. Write a java program that implements the multithreaded application that has four threads. First thread generates ‘n’ number of random integers (user defined), the time elapse between each random generation should be 2 seconds, if the generated number is odd, then second thread computes the cube of that number and prints it, if the number generated by first thread is even then third thread computes the square of the number and prints it. Now, Fourth thread has to sum up all the generated numbers and prints it.

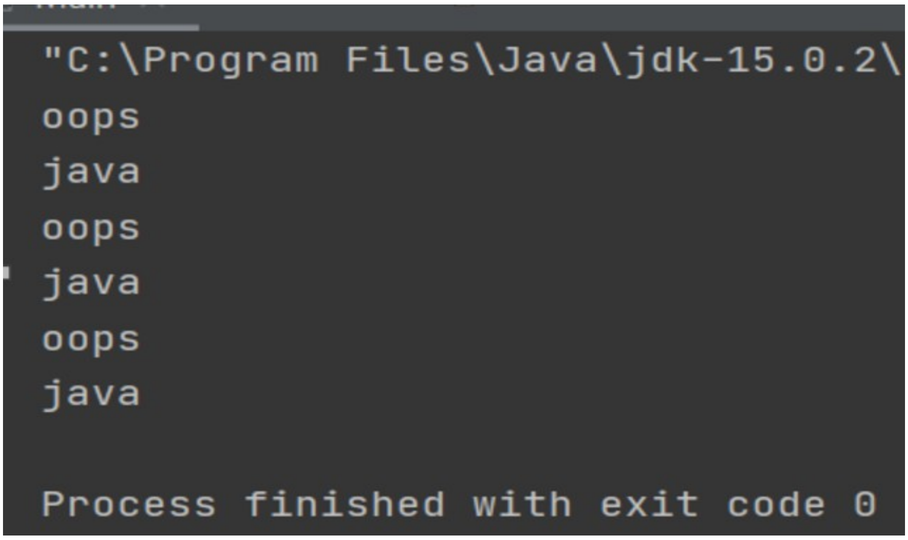
```
package com.deepak;
import java.util.*;
class SquareThread extends Thread {
    int number;

    SquareThread(int randomNumbern) {
        number = randomNumbern;
    }

    public void run() {
        System.out.println("Square of " + number + " = " + (number * number));
    }
}

class CubeThread extends Thread {
    int number;

    CubeThread(int randomNumber) {
        number = randomNumber;
    }

    public void run() {
        System.out.println("Cube of " + number + " = " + number * number * number);
    }
}
class SumThread extends Thread
{
    int sum=0;
    SumThread(int arr[])
    {
        for(int i=0;i<arr.length;i++)
            sum+=arr[i];
    }
    public void run() {
        System.out.println("Sum of all the generated random integers:" + sum);
    }
}
class RandomNumberThread extends Thread {
    public void run() {
        Scanner inp = new Scanner(System.in);
        System.out.print("Enter number of random integers: ");
        int n=inp.nextInt();
        int arr[]= new int[n];
        Random random = new Random();
        for (int i = 0; i < n; i++) {
            int randomInteger = random.nextInt(100);
            arr[i]=randomInteger;
            System.out.println("Random Integer generated : " + randomInteger);
            if((randomInteger%2) == 0) {
                SquareThread sThread = new SquareThread(randomInteger);
                sThread.start();
            }
            else {
                CubeThread cThread = new CubeThread(randomInteger);
                cThread.start();
            }
            try {
                Thread.sleep(2000);
            }
            catch (InterruptedException ex) {
                System.out.println(ex);
            }
        }
        SumThread sumTh = new SumThread(arr);
        sumTh.start();
    }
}

public class Main {
    public static void main(String args[]) {
        RandomNumberThread rnThread = new RandomNumberThread();
        rnThread.start();
    }
}
```



```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe"
Enter number of random integers: 3
Random Integer generated : 21
Cube of 21 = 9261
Random Integer generated : 97
Cube of 97 = 912673
Random Integer generated : 89
Cube of 89 = 704969
Sum of all the generated random integers:207

Process finished with exit code 0
```

10. Write a multithreaded program that continuously prints the strings “OOPS” and “JAVA” in the console at random in the time elapse of 2 seconds. Use two threads one for “OOPS” another for “JAVA”.

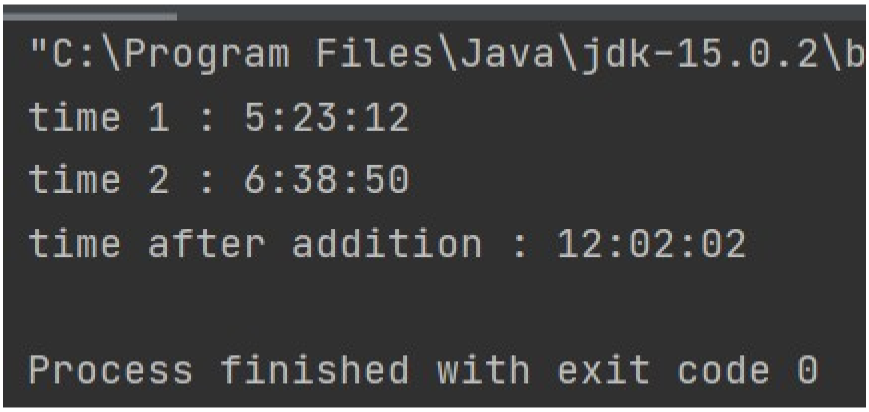
```
package com.deepak;
import java.util.Scanner;
import java.lang.*;
import java.io.*;

class mythread extends Thread
{

    public void run()
    {
        for (int i=0;i<3;i++) {
            try{
                Thread.sleep(2000);
            }
            catch (Exception e)
            {
                System.out.println(e.getMessage());
            }
            System.out.println(Thread.currentThread().getName());

        }
    }
}

public class Main {

    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        mythread t1=new mythread();
        mythread t2=new mythread();
        t1.setName("java");
        t2.setName("oops");
        t1.start();
        t2.start();
    }
}
```



```
"C:\Program Files\Java\jdk-15.0.2\
oops
java
oops
java
oops
java

Process finished with exit code 0
```

11. Create a class called as clock which has three instance variables hours, minutes and seconds. It contains method called as add() which takes clock object as parameter and sets the data in the body and return the object. Now in main method create two clock objects and each one passes 3 arguments via parameterized constructor. Now create third object and this third object is assigned with adding the time values of each two objects Like hint : Clock c3 = c1.add(c2) Now after this print the addition of two time values Testcases : Time 1 : 5:23:12 Time 2: 6:38:50 Time after addition : 12:2:2

```
package com.deepak;
import java.util.Scanner;
import java.lang.*;
import java.io.*;

class clock{
    int hh,mm,ss;
    clock(int h, int m, int s)
    {
        hh=h;
        mm=m;
        ss=s;
    }
    void show_time()
    {
        System.out.print(hh+":");
        if (mm<10)
            System.out.print("0"+mm+":");
        else
            System.out.print(mm+":");
        if (ss<10)
            System.out.print("0"+ss);
        else
            System.out.print(ss);
        System.out.println();
    }

    clock add(clock c)
    {
        int s,carry,m,h ;
        s=c.ss+this.ss;
        carry=s/60;
        s=s%60;
        m=c.mm+this.mm+carry;
        carry=m/60;
        m=m%60;
        h=c.hh+this.hh+carry;
        return new clock(h,m,s);
    }
}

public class Main {

    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        clock c1=new clock(5,23,12);
        clock c2=new clock(6,38,50);
        clock c3=c1.add(c2);
        System.out.print("time 1 : ");
        c1.show_time();
        System.out.print("time 2 : ");
        c2.show_time();
        System.out.print("time after addition : ");
        c3.show_time();
    }
}
```



```
"C:\Program Files\Java\jdk-15.0.2\b
time 1 : 5:23:12
time 2 : 6:38:50
time after addition : 12:02:02

Process finished with exit code 0
```