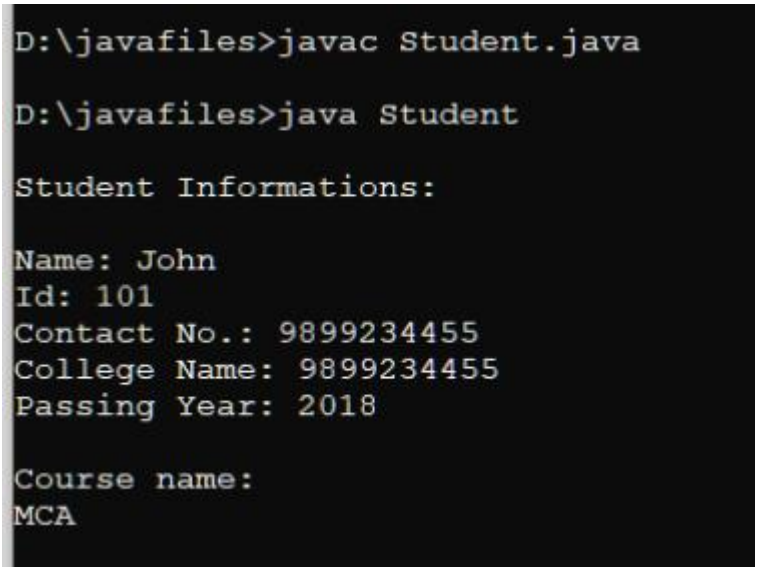
PROGRAM:

```
Public class Student
{
int id, passout_Year;
String name, contact_No, college_Name;
//constructor
Student (String contact_No, String college_Name, int passout_Year)
{
this.contact_No = contact_No;
this.college_Name = college_Name;
this.passout_Year = passout_Year;
}
//constructor oveloading
Student (int id, String name)
 {
this ("9899234455", "IIT Kanpur", 2018);
this.id = id;
this.name = name;
}
//method
void display ()
{
System.out.println("\nCourse name:");
}

//method overloading
void display (String course_name)
{
System.out.println(course_name);
}
public static void main (String args[])
{
//object creation
Student s=new Student (101, "John");
System.out.println("\nStudent Informations: \n");
System.out.println("Name: "+s.name+"\nId: "+s.id+"\nContact No.:
"+s.contact_No+"\nCollege Name: "+s.college_Name+"\nPassing Year:
"+s.passout_Year);
s.display();
s.display("MCA");
}
}
```

OUTPUT:


D:\javafiles>javac Student.java
D:\javafiles>java Student
Student Informations:
Name: John
Id: 101
Contact No.: 9899234455
College Name: 9899234455
Passing Year: 2018
Course name:
MCA

PROGRAM:

```
import java.io.*;
class Student
  {
        int roll_number; void
        getValue(int n)
        {
            roll_number=n;
        }
        void putValue( )
         {
            System.out.println("Roll No.:"+roll_number);
         }
    }
class Test extends Student
   {
         float part1,part2;
         void getValue(float m1, float m2)
          {
              part1 = m1;
              part2 = m2;
           }
          void putMarks( )
           {
             System.out.println("MARKS OBTAINED");
             System.out.println("Mark 1="+part1);
             System.out.println("Mark 2="+part2);
            }
}
interface Sports
  {
      float swt=6.0F;
      void putwt();
   }
class Results extends Test implements Sports
{
    float total;
    public void putwt( )
     {
       System.out.println("Sportswt="+swt);
      }
```

```
    void display( )
     {
        total=part1+part2+swt;
        putValue(  ); putMarks( );
        putwt( );
        System.out.println("Total Score="+total);
     }
  }
class Output
{
    public static void main(String args[ ])
     {
        Results student1=new Results();
        student1.getValue(1234);
        student1.getValue(27.5F,33.0F);
        student1.display();
     }
}
```

OUTPUT:

```
E:\JAVA>javac Output.java

E:\JAVA>java Output
Roll No.:1
MARKS OBTAINED
Mark 1=80.5
Mark 2=77.0
Sportswt=66.0
Total Score=223.5
```

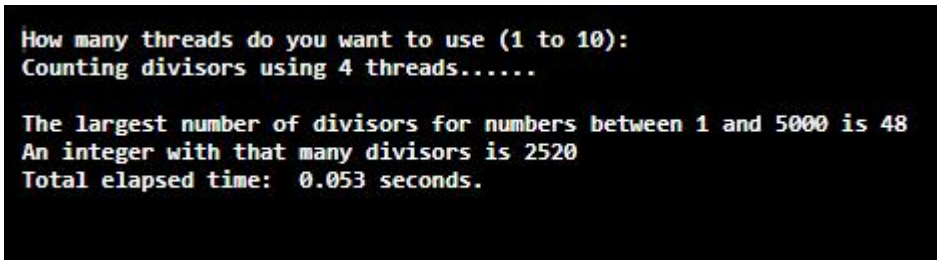
PROGRAM:

```
import java.util.Scanner;
public class Main
{
   private final static int MAX = 5000;
   private volatile static int Divisor = 0;
   private volatile static int MaxDivisorCount;
   synchronized private static void report(int maxCountFromThread, int
intWithMaxFromThread)
   {
      if (maxCountFromThread > Divisor)
      {
         Divisor = maxCountFromThread;
         MaxDivisorCount = intWithMaxFromThread;
      }
   }

   private static class CountDivisorsThread extends Thread
   {
   int min, max;
   public CountDivisorsThread(int min, int max)
   {
         this.min = min;
         this.max = max;
   }
   public void run()
   {
   int maxDivisors = 0;
   int whichInt = 0;
   for (int i = min; i < max; i++)
   {
   int divisors = countDivisors(i);
   if (divisors > maxDivisors)
   {
    maxDivisors = divisors;
    whichInt = i;
    }
    }
    report(maxDivisors,whichInt);
   }
   }

   private static void countDivisorsWithThreads(int numberOfThreads) {
      System.out.println("\nCounting divisors using " + numberOfThreads + "
threads.....");
      long startTime = System.currentTimeMillis();
```

```
CountDivisorsThread[] worker = new CountDivisorsThread[numberOfThreads];
      int integersPerThread = MAX/numberOfThreads;
      int start = 1;
      int end = start + integersPerThread - 1;
      for (int i = 0; i < numberOfThreads; i++) {
        if (i == numberOfThreads - 1) {
          end = MAX;
        }
        worker[i] = new CountDivisorsThread( start, end );
        start = end+1;
        end = start + integersPerThread - 1;
      }
      Divisor = 0;
      for (int i = 0; i < numberOfThreads; i++)
        worker[i].start();
      for (int i = 0; i < numberOfThreads; i++) {

        while (worker[i].isAlive()) {
          try {
            worker[i].join();
          }
          catch (InterruptedException e) {
          }
        }
      }
      long elapsedTime = System.currentTimeMillis() - startTime;
      System.out.println("\nThe largest number of divisors " +  "for numbers between
1 and " + MAX + " is " + Divisor);
      System.out.println("An integer with that many divisors is " + MaxDivisorCount);
      System.out.println("Total elapsed time:  " + (elapsedTime/1000.0) + "
seconds.\n");
    }

    public static void main(String[] args) {
      Scanner in = new Scanner(System.in);
      int numberOfThreads = 0;
      while (numberOfThreads < 1 || numberOfThreads > 10) {
        System.out.print("How many threads do you want to use (1 to 10): ");
        numberOfThreads = in.nextInt();
        if (numberOfThreads < 1 || numberOfThreads > 10)
          System.out.println("Please enter a number from 1 to 10:");
      }
      countDivisorsWithThreads(numberOfThreads);
    }
```

```
public static int countDivisors(int N) {
      int count = 0;
      for (int i = 1; i <= N ; i++) {
         if ( N % i == 0 )
            count ++;
      }
      return count;
   }

}
```

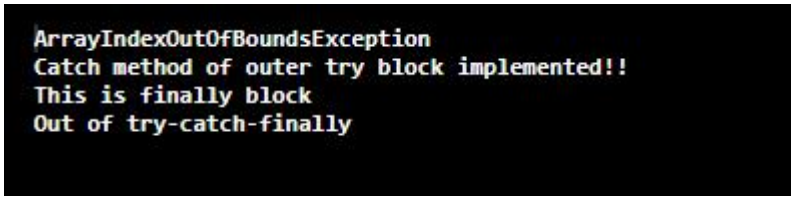
OUTPUT:



PROGRAM:

```
import java.io.*;
public class TestClass {
public static void main(String args[])
{
// Outer try block
try {
int a[] = { 1, 2, 3, 4, 5, 6, 7 };
// printing element at index 7
System.out.println(a[7]);
// inner try block
try {
// division by zero
int x = a[2] / 0;
}
catch (ArithmeticException e2)
{
System.out.println("Number cannot be divided by zero");
}}
catch (ArrayIndexOutOfBoundsException e1)
{
System.out.println("ArrayIndexOutOfBoundsException");
System.out.println("Catch method of outer try block implemented!!");
}
finally{
        System.out.println("This is finally block");
    }
    System.out.println("Out of try-catch-finally");
}}
```

OUTPUT:



PROGRAM:

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
public class ReadFile {
  public static void main(String[] args) {
    try {
      File myObj = new File("sample.txt");
      Scanner myReader = new Scanner(myObj);
      while (myReader.hasNextLine()) {
        String data = myReader.nextLine();
        System.out.println(data);
      }
      myReader.close();
    } catch (FileNotFoundException e) {
      System.out.println("An error occurred.");
      e.printStackTrace();
    }
  }
}
```

OUTPUT:

PROGRAM:

ClassOne.java

```
package package_name;
public class ClassOne {
public void methodClassOne() {
System.out.println("Hello there its ClassOne");
}
}
```

ClassTwo.java

```
package package_one;
public class ClassTwo {
public void methodClassTwo(){
System.out.println("Hello there i am ClassTwo");
}
}
```

Testing.java

```
import package_one.ClassTwo;
import package_name.ClassOne;

public class Testing {
public static void main(String[] args){
ClassTwo a = new ClassTwo();
ClassOne b = new ClassOne();
a.methodClassTwo();
b.methodClassOne();
}
}
```

OUTPUT:

For creating packages

```
C:\jdk-16.0.2\bin>javac -d . ClassOne.java

C:\jdk-16.0.2\bin>javac -d . ClassTwo.java
```

Output

```
C:\jdk-16.0.2\bin>javac Testing.java

C:\jdk-16.0.2\bin>java Testing.java
Hello there i am ClassTwo
Hello there its ClassOne
```