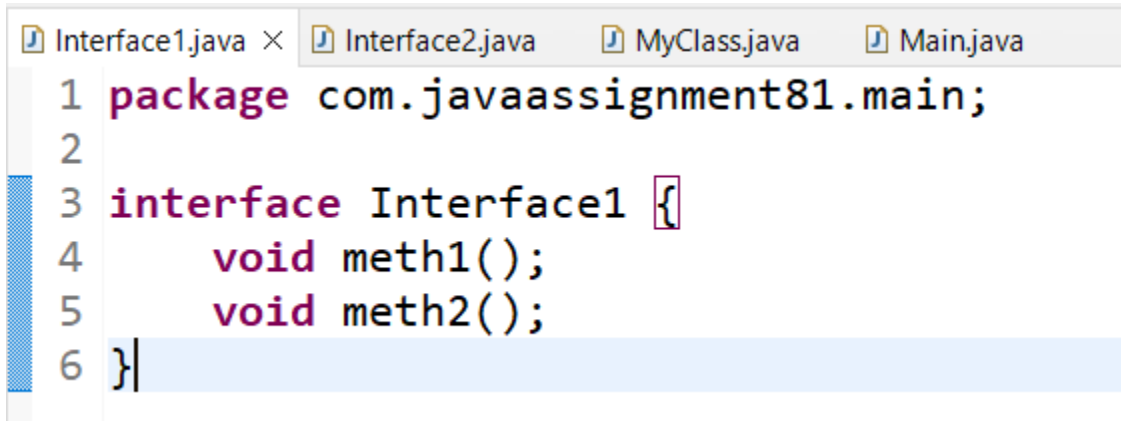
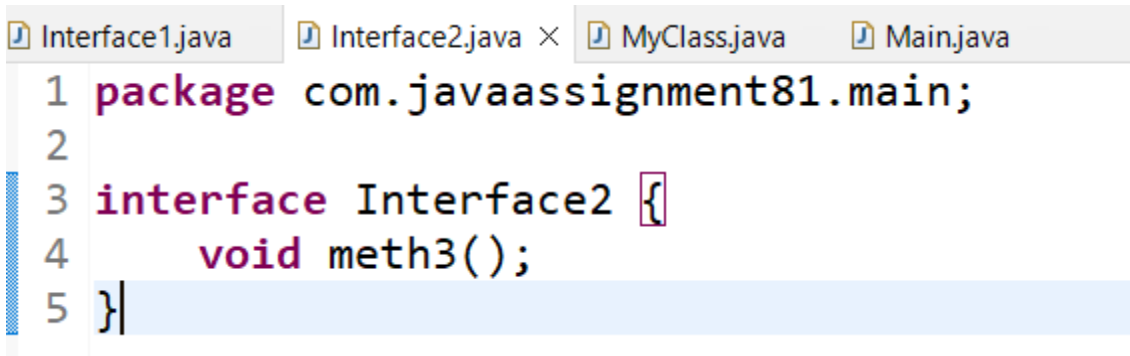


**PG DAC–March 2023**  
**C-DAC THIRUVANANTHAPURAM**  
**JAVA- LAB 8**

1. Create one interface with two methods meth1(), meth2() , also create one more interface with a method meth3(). Try to implement the two interfaces in a class and invoke the function in main().



```
Interface1.java × Interface2.java MyClass.java Main.java
1 package com.javaassignment81.main;
2
3 interface Interface1 {
4     void meth1();
5     void meth2();
6 }
```



```
Interface1.java Interface2.java × MyClass.java Main.java
1 package com.javaassignment81.main;
2
3 interface Interface2 {
4     void meth3();
5 }
```

```
Interface1.java  Interface2.java  MyClass.java ×  Main.java
1 package com.javaassignment81.main;
2
3 class MyClass implements Interface1, Interface2 {
4     public void meth1() {
5         System.out.println("Inside meth1");
6     }
7
8     public void meth2() {
9         System.out.println("Inside meth2");
10    }
11
12    public void meth3() {
13        System.out.println("Inside meth3");
14    }
15 }
```

```
Interface1.java  Interface2.java  MyClass.java  Main.java ×  Console ×
1 package com.javaassignment81.main;
2
3 public class Main {
4     public static void main(String[] args) {
5         MyClass obj = new MyClass();
6         obj.meth1();
7         obj.meth2();
8         obj.meth3();
9     }
10 }
11 |
```

<terminated> Main (1) [Java Application]  
Inside meth1  
Inside meth2  
Inside meth3

2. Write an interface called Exam with a method Pass(int mark) that returns a boolean value. Write another interface called Classify with a method Division(int average) which returns a string. Write a class called Result which implements both Exam and Classify.

The Pass method should return true if the mark is greater than or equal to 50 else false. The Division method must return "First" when the parameter average is 60 or more, " Second" when average is 50 or more but below 60, "No Division" when average is less than 50.

```
Exam.java × Classify.java Result.java Main.java
1 package com.javaassignment82.main;
2
3 interface Exam {
4     boolean Pass(int mark);
5 }
6
```

```
Exam.java Classify.java × Result.java Main.java
1 package com.javaassignment82.main;
2
3 interface Classify {
4     String Division(int average);
5 }
```

```
Exam.java Classify.java Result.java X Main.java
1 package com.javaassignment82.main;
2
3 class Result implements Exam, Classify {
4     public boolean Pass(int mark) {
5         return mark >= 50;
6     }
7
8     public String Division(int average) {
9         if (average >= 60) {
10             return "First";
11         } else if (average >= 50) {
12             return "Second";
13         } else {
14             return "No Division";
15         }
16     }
17 }
```

```
Exam.java X Classify.java Result.java Main.java X
1 package com.javaassignment82.main;
2
3 public class Main {
4     public static void main(String[] args) {
5         Result result = new Result();
6         int mark = 65;
7         int average = 70;
8         boolean pass = result.Pass(mark);
9         String division = result.Division(average);
10        System.out.println("Pass: " + pass);
11        System.out.println("Division: " + division);
12    }
13 }
```

```
Console X
<terminated> Main (2) [Java Applicati
Pass: true
Division: First
```

3. Create two interfaces with a method having same name and signature, also implement the two interfaces in a class and invoke that function with class instance.

Interface1.java × Interface2.java Main.java Myname.java

```
1 package com.javaassignment83.main;
2
3 interface Interface1 {
4     public void methodName();
5     public void methodSignature();
6 }
7
```

Interface1.java × Interface2.java × Main.java Myname.java

```
1 package com.javaassignment83.main;
2
3 interface Interface2 {
4     public void methodName();
5     public void methodSignature();
6 }
7
```

Interface1.java Interface2.java Myname.java × Main.java

```
1 package com.javaassignment83.main;
2
3 class Myname implements Interface1, Interface2 {
4     public void methodName() {
5         System.out.println("My Name is Monika Srivastava!");
6     }
7     public void methodSignature() {
8         System.out.println("My Signature is Monika Srivastava");
9     }
10 }
11
```

```

1 package com.javaassignment83.main;
2
3 public class Main {
4     public static void main(String[] args) {
5         Myname obj = new Myname();
6         obj.methodName();
7         Myname obj1 = new Myname();
8         obj1.methodSignature();
9     }
10 }
11

```

Console Output:

```

<terminated> Main (3) [Java Application] D:\Eclipse\ eclipse\plugin
My Name is Monika Srivastava!
My Signature is Monika Srivastava

```

4. Write an interface called Numbers with a method `int Process(int x, int y)`. Write a class called Sum, in which the method Process finds the sum of two numbers and returns an integer value. Write another class called average, in which the Process method finds the average of two numbers and returns a float value.

```

1 package com.javaassignment84.main;
2
3 interface Numbers {
4     public int Process(int x, int y);
5 }
6

```

```

1 package com.javaassignment84.main;
2
3 class Sum implements Numbers {
4     public int Process(int x, int y) {
5         return x + y;
6     }
7 }

```

```

Numbers.java Sum.java Average.java × Main.java
1 package com.javaassignment84.main;
2
3 class Average implements Numbers {
4     public int Process(int x, int y) {
5         return (int)(x + y) / 2;
6     }
7 }
8

```

```

Numbers.java Sum.java Average.java Main.java × Console ×
1 package com.javaassignment84.main;
2
3 public class Main {
4     public static void main(String[] args) {
5         Sum s = new Sum();
6         Average a = new Average();
7
8         int result1 = s.Process(15, 10);
9         float result2 = a.Process(10, 30);
10
11         System.out.println("Sum: " + result1);
12         System.out.println("Average: " + result2);
13     }
14 }
15

```

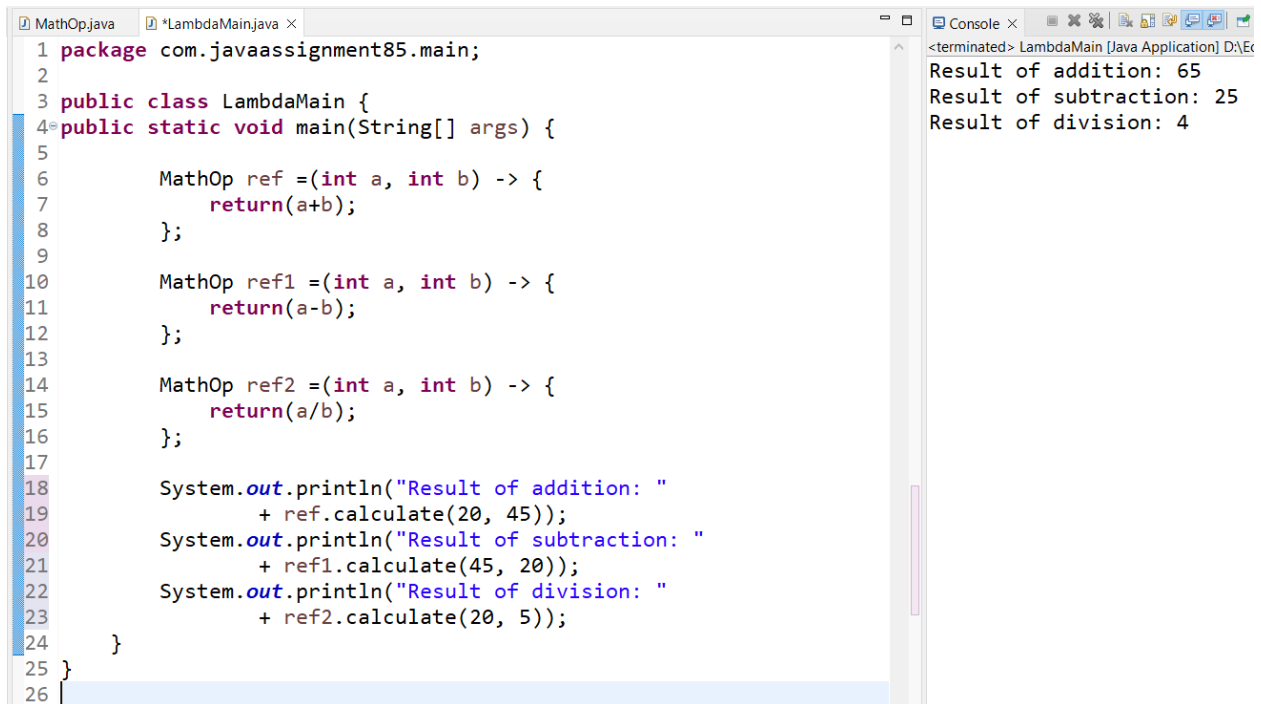
<terminated> Main (4) [Java Applic  
Sum: 25  
Average: 20.0

5. Write a java program to do addition subtraction and division using lambda Expression.

```

MathOp.java × *LambdaMain.java
1 package com.javaassignment85.main;
2
3 public interface MathOp {
4
5     int calculate(int x, int y);
6
7 }
8

```



```

1 package com.javaassignment85.main;
2
3 public class LambdaMain {
4     public static void main(String[] args) {
5
6         MathOp ref =(int a, int b) -> {
7             return(a+b);
8         };
9
10        MathOp ref1 =(int a, int b) -> {
11            return(a-b);
12        };
13
14        MathOp ref2 =(int a, int b) -> {
15            return(a/b);
16        };
17
18        System.out.println("Result of addition: "
19            + ref.calculate(20, 45));
20        System.out.println("Result of subtraction: "
21            + ref1.calculate(45, 20));
22        System.out.println("Result of division: "
23            + ref2.calculate(20, 5));
24    }
25 }
26

```

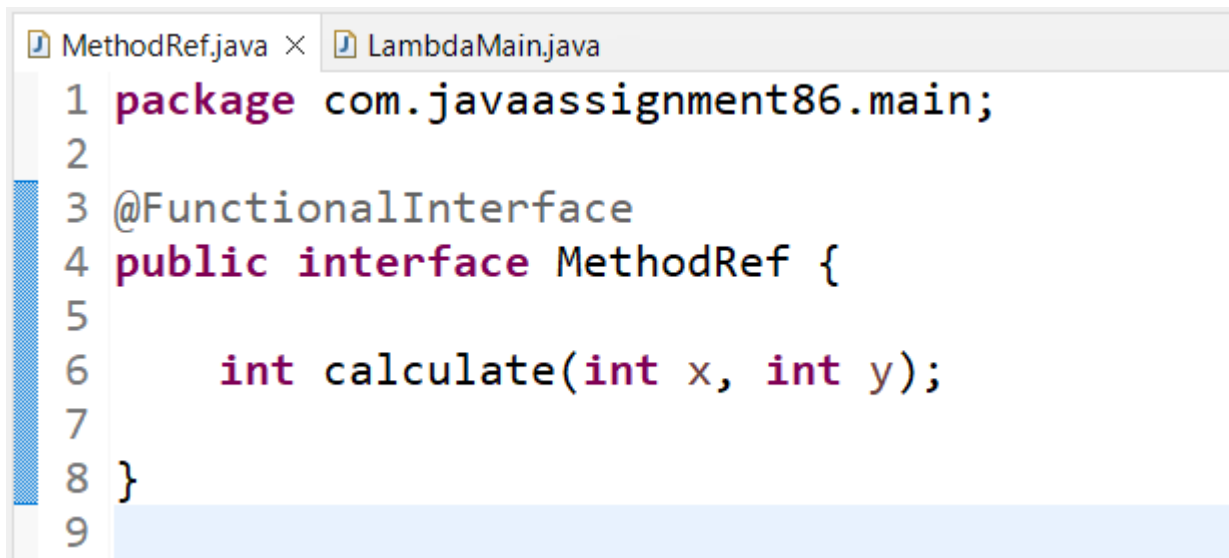
Console Output:

```

<terminated> LambdaMain [Java Application] D:\E
Result of addition: 65
Result of subtraction: 25
Result of division: 4

```

**6. Write a java program to find sum and product of two numbers using lambda and function interface  
(Use method reference)**

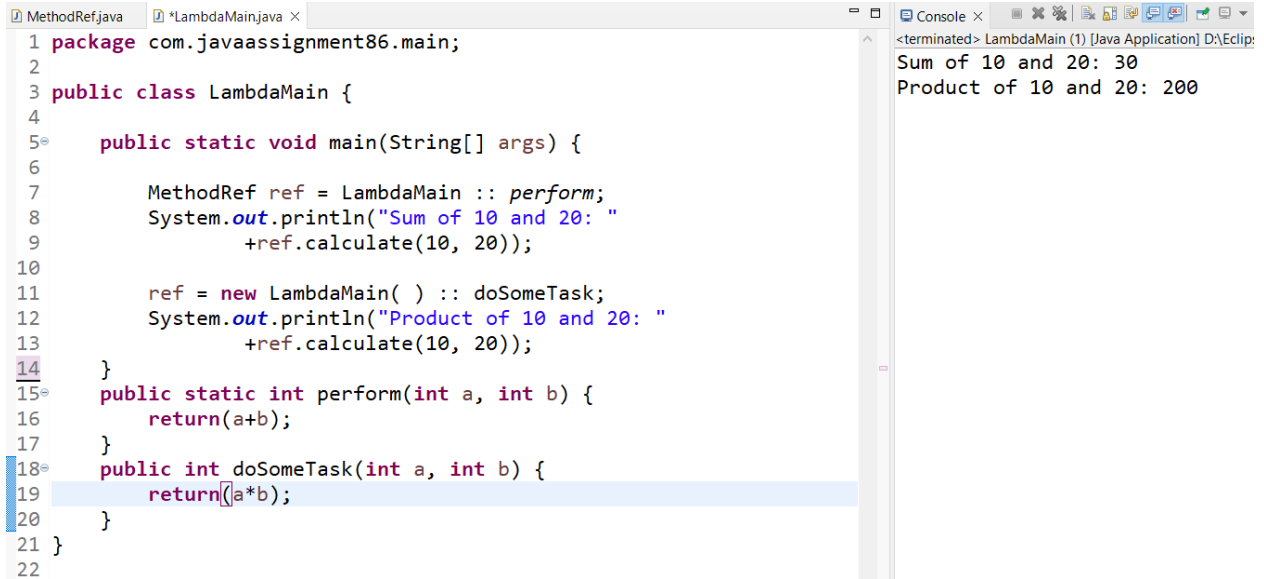


```

1 package com.javaassignment86.main;
2
3 @FunctionalInterface
4 public interface MethodRef {
5
6     int calculate(int x, int y);
7
8 }
9

```





```
1 package com.javaassignment86.main;
2
3 public class LambdaMain {
4
5     public static void main(String[] args) {
6
7         MethodRef ref = LambdaMain :: perform;
8         System.out.println("Sum of 10 and 20: "
9             +ref.calculate(10, 20));
10
11         ref = new LambdaMain( ) :: doSomeTask;
12         System.out.println("Product of 10 and 20: "
13             +ref.calculate(10, 20));
14     }
15     public static int perform(int a, int b) {
16         return(a+b);
17     }
18     public int doSomeTask(int a, int b) {
19         return(a*b);
20     }
21 }
22
```

<terminated> LambdaMain (1) [Java Application] D:\Eclip:  
Sum of 10 and 20: 30  
Product of 10 and 20: 200

7. Create three classes Faculty (facultyid, salary) FullTimeFaculty (basic, allowance) inherits class Faculty Part Time Faculty (hour, rate) inherits class Faculty Create a method for accepting input in FullTimeFaculty and PartTimeFaculty, but salary should not be accepted. Salary is calculated on the basis of (basic+allowance) for FullTimeFaculty and (hour\*rate) for PartTimeFaculty. Also create method in above classes to display faculty data. Create another class(say XYZ) for main method and store 2 fulltime and 2 parttime faculty information. Also print their details.

```
*Faculty.java × FullTimeFaculty.java PartTimeFaculty.java *Main.java
1 package com.javaassignment87.main;
2
3 class Faculty {
4     private int facultyId;
5     private double salary;
6
7     public Faculty(int facultyId, double salary) {
8         this.setFacultyId(facultyId);
9         this.setSalary(salary);
10    }
11    public int getFacultyId() {
12        return facultyId;
13    }
14    public double getSalary() {
15        return salary;
16    }
17    public void display() {
18        System.out.println("Faculty ID: " + getFacultyId());
19        System.out.println("Salary: " + getSalary());
20    }
21    public void setFacultyId(int facultyId) {
22        this.facultyId = facultyId;
23    }
24    public void setSalary(double salary) {
25        this.salary = salary;
26    }
27 }
28
```

```
*Faculty.java × *FullTimeFaculty.java × PartTimeFaculty.java *Main.java
1 package com.javaassignment87.main;
2 import java.util.Scanner;
3
4 class FullTimeFaculty extends Faculty {
5     private double basic;
6     private double allowance;
7
8     public FullTimeFaculty(int facultyId, double basic,
9         double allowance) {
10         super(facultyId, 0);
11         this.basic = basic;
12         this.allowance = allowance;
13         calculateSalary();
14     }
15     private void calculateSalary() {
16         setSalary(basic + allowance);
17     }
18     public void acceptInput() {
19         Scanner sc = new Scanner(System.in);
20         System.out.print("Enter faculty ID: ");
21         setFacultyId(sc.nextInt());
22         System.out.print("Enter basic salary: ");
23         basic = sc.nextDouble();
24         System.out.print("Enter allowance: ");
25         allowance = sc.nextDouble();
26         calculateSalary();
27     }
28     public void display() {
29         super.display();
30         System.out.println("Basic salary: " + basic);
31         System.out.println("Allowance: " + allowance);
32     }
33 }
```

```
*Faculty.java × *FullTimeFaculty.java *PartTimeFaculty.java × *Main.java
1 package com.javaassignment87.main;
2 import java.util.Scanner;
3
4 class PartTimeFaculty extends Faculty {
5     private double hour;
6     private double rate;
7
8     public PartTimeFaculty(int facultyId, double hour, double rate)
9     { super(facultyId, 0);
10         this.hour = hour;
11         this.rate = rate;
12         calculateSalary();
13     }
14     private void calculateSalary() {
15         setSalary(hour * rate);
16     }
17     public void acceptInput() {
18         Scanner sc = new Scanner(System.in);
19         System.out.print("Enter faculty ID: ");
20         setFacultyId(sc.nextInt());
21         System.out.print("Enter number of hours: ");
22         hour = sc.nextDouble();
23         System.out.print("Enter hourly rate: ");
24         rate = sc.nextDouble();
25         calculateSalary();
26     }
27     public void display() {
28         super.display();
29         System.out.println("Number of hours: " + hour);
30         System.out.println("Hourly rate: " + rate);
31     }
32 }
```

```

1 package com.javaassignment87.main;
2
3 public class Main {
4     public static void main(String[] args) {
5         FullTimeFaculty ftf1 = new FullTimeFaculty(101, 50000, 10000);
6         PartTimeFaculty ptf1 = new PartTimeFaculty(201, 20, 1000);
7
8         ftf1.acceptInput();
9
10        System.out.println("Full-time faculty details:");
11        ftf1.display();
12
13        ptf1.acceptInput();
14
15        System.out.println("Part-time faculty details:");
16        ptf1.display();
17    }
18 }
19

```

Console Output:

```

<terminated> Main (5) [Java Application] D:\Eclipse\ eclipse\plugins
Enter faculty ID: 101
Enter basic salary: 10000
Enter allowance: 2000
Full-time faculty details:
Faculty ID: 101
Salary: 12000.0
Basic salary: 10000.0
Allowance: 2000.0
Enter faculty ID: 201
Enter number of hours: 10
Enter hourly rate: 500
Part-time faculty details:
Faculty ID: 201
Salary: 5000.0
Number of hours: 10.0
Hourly rate: 500.0

```

8. Create an Abstract class Processor with int member variable data and method showData to display data value. a.Create abstract method process() to define processing of member data.a. Create a class Factorial using abstract class Processor to calculate and print factorial of a number by overriding the process method. b. Create a class Circle using abstract class Processor to calculate and print area of a circle by overriding the process method Ask user to enter choice (factorial or circle area). Also ask data to work upon; Use Processor class reference to achieve this mechanism

```

1 package com.javaassignment88.main;
2
3 abstract class Processor {
4     int data;
5
6     void showData() {
7         System.out.println("Value of Data: " + data);
8     }
9
10    abstract void process();
11 }
12
13

```

```

1 package com.javaassignment88.main;
2
3 class Factorial extends Processor {
4     void process() {
5         int fact = 1;
6         for (int i = 1; i <= data; i++) {
7             fact *= i;
8         }
9         System.out.println("Factorial of " + data + " is " + fact);
10    }
11 }
12

```

```

<terminated> Main (6) [Java Application] D:\Eclipse\ eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.j2se-17.0.2\jre\bin\java.exe
Enter choice (1 for factorial, 2 for circle area):
1
Enter data:
10
Value of Data: 10
Factorial of 10 is 3628800

```

```

1 package com.javaassignment88.main;
2
3 class Circle extends Processor {
4     void process() {
5         double area = Math.PI * data * data;
6         System.out.println("Area of circle with radius " + data + " is " + area);
7     }
8 }
9

```

```

<terminated> Main (6) [Java Application] D:\Eclipse\ eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.j2se-17.0.2\jre\bin\java.exe
Enter choice (1 for factorial, 2 for circle area):
2
Enter data:
20
Value of Data: 20
Area of circle with radius 20 is 1256.6370614359173

```

```

1 package com.javaassignment88.main;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         System.out.println("Enter choice (1 for factorial, "
9             + "2 for circle area):");
10        int choice = sc.nextInt();
11
12        Processor processor;
13        if (choice == 1) {
14            processor = new Factorial();
15        } else if (choice == 2) {
16            processor = new Circle();
17        } else {
18            System.out.println("Your choice is invalid");
19            return;
20        }
21
22        System.out.println("Enter data:");
23        processor.data = sc.nextInt();
24
25        processor.showData();
26        processor.process();
27    }
28 }

```

```

<terminated> Main (6) [Java Application] D:\Eclipse\ eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.j2se-17.0.2\jre\bin\java.exe
Enter choice (1 for factorial, 2 for circle area):
10
Your choice is invalid

```