

**NAME: Pratham Srivastava**

**Roll No: 2401330130192**

**Branch: IT**

**Section: C**

**Workshop Lab Number : D-203**

# Problem-1: SOURCE CODE

## 1. Main Class-

```
2. import java.util.*;
3.
4. class Employee {
5.     private String name;
6.     private int age;
7.     private double salary;
8.
9.     public Employee(String name, int age, double salary) {
10.         this.name = name;
11.         this.age = age;
12.         this.salary = salary;
13.     }
14.
15.     public String getName() {
16.         return name;
17.     }
18.
19.     public void setName(String name) {
20.         this.name = name;
21.     }
22.
23.     public int getAge() {
24.         return age;
25.     }
26.
27.     public void setAge(int age) {
28.         this.age = age;
29.     }
30.
31.     public double getSalary() {
32.         return salary;
33.     }
34.
35.     public void setSalary(double salary) {
36.         this.salary = salary;
37.     }
38.
39. }
40.
41. public class Problem1 {
42.     public static void main(String[] args) {
43.         List<Employee> employees = new ArrayList<>();
44.         employees.add(new Employee("Random1", 30, 15000));
45.         employees.add(new Employee("Random2", 25, 20000));
46.         employees.add(new Employee("Random3", 40, 8000));
```

```

47.     employees.add(new Employee("Random4", 21, 45600));
48.
49.     System.out.println("Before sorting:");
50.     for (Employee employee : employees) {
51.         System.out.println(employee.getSalary());
52.     }
53.
54.     Collections.sort(employees, (e1, e2) ->
Double.compare(e2.getSalary(), e1.getSalary()));
55.
56.     System.out.println("\nAfter sorting:");
57.     for (Employee employee : employees) {
58.         System.out.println(employee.getSalary());
59.     }
60. }
61. }

```

## Problem-1: OUTPUT

The screenshot shows an IDE with the following components:

- EXPLORER:** A list of files including `Sessional-2(OOTs)`, `com`, `~$athamSrivastava_IT_C.docx`, `J Employee.class`, `J NegativeNumberNotAllowedException.class`, `J PrathamSrivastava_IT_C.docx`, `J Problem1.class`, `J Problem1.java`, `J Problem4.class`, `J Problem4.java`, `J Problem5.class`, and `J Problem5.java`.
- Problem1.java:** The code is identical to the one in the first block, showing the sorting of employees by salary in descending order.
- OUTPUT:** The output of the program is displayed in the terminal window. It shows the salaries of five employees before and after sorting.
 

```

C:\Users\Pankaj Srivastava\Desktop\Sessional-2(OOTs)>javac Problem1.java

C:\Users\Pankaj Srivastava\Desktop\Sessional-2(OOTs)>java Problem1
Before sorting:
15000.0
20000.0
8000.0
45600.0

After sorting:
45600.0
20000.0
15000.0
8000.0

```

## Problem-2: SOURCE CODE

### Main Class –

```
package com.main;  
  
import com.maths.*;  
  
import com.maths.operations.Calculator;  
  
public class Main {  
    public static void main(String[] args) {  
        Calculator calculator = new Calculator();  
  
        int result1 = calculator.add(5, 3);  
        System.out.println("5 + 3 = " + result1);  
  
        int result2 = calculator.subtract(7, 4);  
        System.out.println("7 - 4 = " + result2);  
  
        int result3 = calculator.multiply(8, 2);  
        System.out.println("8 * 2 = " + result3);  
  
        double result4 = calculator.divide(16, 4);  
        System.out.println("16 / 4 = " + result4);  
    }  
}
```

## MathOperations Class –

```
package com.maths;
```

```
public class MathOperations {  
    public static int add(int x, int y) {  
        return x + y;  
    }  
  
    public static int subtract(int x, int y) {  
        return x - y;  
    }  
  
    public static int multiply(int x, int y) {  
        return x * y;  
    }  
  
    public static int divide(int x, int y) {  
        return x / y;  
    }  
}
```

## Calculator Class –

```
package com.maths.operations;
```

```
import com.maths.*;
```

```
public class Calculator {
```

```
    public int add(int a, int b) {  
        return MathOperations.add(a, b);  
    }
```

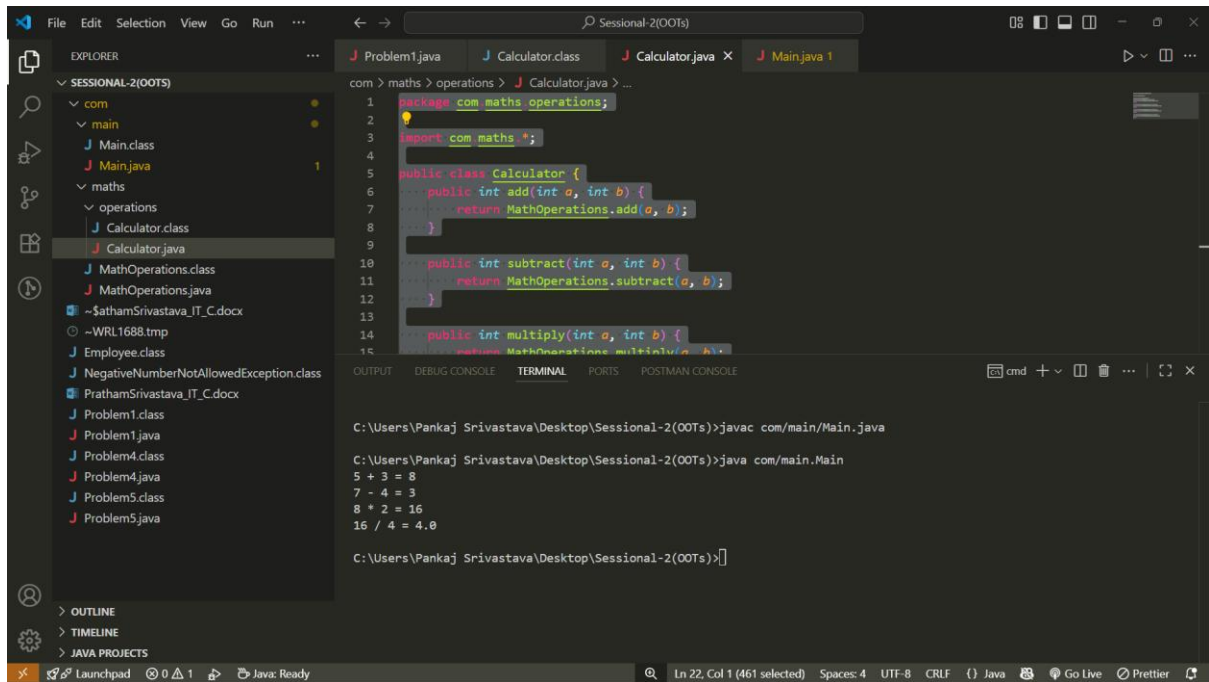
```
    public int subtract(int a, int b) {  
        return MathOperations.subtract(a, b);  
    }
```

```
    public int multiply(int a, int b) {  
        return MathOperations.multiply(a, b);  
    }
```

```
    public double divide(int a, int b) {  
        return MathOperations.divide(a, b);  
    }
```

```
}
```

## Problem-2: OUTPUT



The screenshot displays an IDE with the following components:

- EXPLORER:** Shows a project structure for 'SESSIONAL-2(OOTs)' with packages 'com', 'main', 'maths', and 'operations'. Files include 'Main.class', 'Main.java', 'Calculator.class', 'Calculator.java', 'MathOperations.class', and 'MathOperations.java'.
- Editor:** Displays the code for 'Calculator.java' with the following content:

```
1 package com.maths.operations;
2
3 import com.maths.*;
4
5 public class Calculator {
6     public int add(int a, int b) {
7         return MathOperations.add(a, b);
8     }
9
10    public int subtract(int a, int b) {
11        return MathOperations.subtract(a, b);
12    }
13
14    public int multiply(int a, int b) {
15        return MathOperations.multiply(a, b);
16    }
17 }
```
- TERMINAL:** Shows the execution of the program:

```
C:\Users\Pankaj Srivastava\Desktop\Sessional-2(OOTs)>javac com/main/Main.java
C:\Users\Pankaj Srivastava\Desktop\Sessional-2(OOTs)>java com/main.Main
5 + 3 = 8
7 - 4 = 3
8 * 2 = 16
16 / 4 = 4.0
C:\Users\Pankaj Srivastava\Desktop\Sessional-2(OOTs)>]
```

## Problem-4: SOURCE CODE

```
public class Problem4 {  
    public static void main(String[] args) {  
        String test = "Java ProgRaMMing";  
        String forConcat = "Concatenated";  
  
        String sub = test.substring(0, 7);  
        String con = test.concat(forConcat);  
        int length = test.length();  
  
        String forEquals = "Concatenated";  
        boolean checkEqual = forConcat.equals(forEquals);  
  
        boolean forContains = test.contains("P");  
  
        System.out.println("This is the substring: " + sub);  
        System.out.println("This is the concatenated string: " + con);  
        System.out.println("The length of the string is: " + length);  
  
        System.out.println();  
        System.out.println("String 1: " + forConcat + " String 2: " +  
forEquals);  
        if (checkEqual)  
            System.out.println("Strings are equal.");  
    }  
}
```



```
System.out.println("String are not equal.");

System.out.println();

System.out.println("String to check for (P): " + test);

if (forContains)

    System.out.println("The string contains (P)");

else

    System.out.println("The string does not contain (P)");
```

The screenshot shows an IDE with the following components:

- Explorer Panel:** Lists files in the `SESSIONAL-2(OOTS)` project, including `com`, `~$athamSrivastava_IT_C.docx`, `~WRL1688.tmp`, `Employee.class`, `NegativeNimberNotAllowedException.class`, `PrathamSrivastava_IT_C.docx`, `Problem1.class`, `Problem1.java`, `Problem4.class`, `Problem4.java` (selected), `Problem5.class`, and `Problem5.java`.
- Editor:** Displays the code for `Problem4.java`:
 

```

1  public class Problem4 {
2      public static void main(String[] args) {
3
4
5      }
6
7      System.out.println("This is the substring: " + sub);
8      System.out.println("This is the concatenated string: " + con);
9      System.out.println("The length of the string is: " + length);
10
11      System.out.println();
12      System.out.println("String 1: " + forConcat + " String 2: " + forEquals);
13      if (checkEqual)
14          System.out.println(x:"Strings are equal.");
15      else
16          System.out.println(x:"Strings are not equal.");
17
18      System.out.println();
19  }
20  }
21  }
22  }
23  }
24  }
25  }
26  }

```
- Terminal Panel:** Shows the execution output:
 

```

C:\Users\Pankaj Srivastava\Desktop\Sessional-2(OOTS)>javac Problem4.java

C:\Users\Pankaj Srivastava\Desktop\Sessional-2(OOTS)>java Problem4
This is the substring: Java Pr
This is the concatenated string: Java ProgRaMmingConcatenated
The length of the string is: 16

String 1: Concatenated String 2: Concatenated
Strings are equal.

String to check for (P): Java ProgRaMming
The string contains (P)

C:\Users\Pankaj Srivastava\Desktop\Sessional-2(OOTS)>

```

## Problem-5: SOURCE CODE

```
import java.util.Scanner;

class NegativeNumberNotAllowedException extends Exception {
}

public class Problem5 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int n = 0;
        System.out.print("Enter the size of the array: ");
        n = scanner.nextInt();

        int[] numbers = new int[n];
        int index = 0;

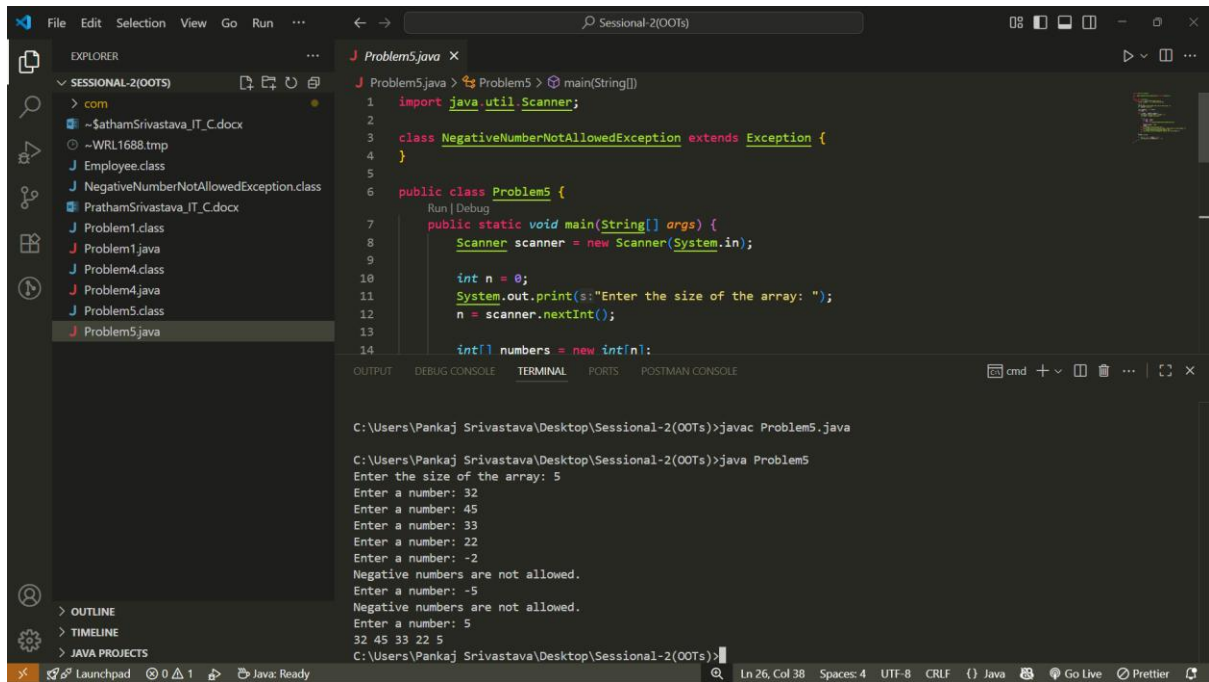
        while (index < numbers.length) {
            System.out.print("Enter a number: ");
            int input = scanner.nextInt();

            try {
                int num = input;
                if (num < 0) {
```

```
        throw new NegativeNumberNotAllowedException();
    }
    numbers[index] = num;
    index++;
} catch (NumberFormatException e) {
    System.out.println("Invalid input, please enter a valid
number.");
} catch (NegativeNumberNotAllowedException e) {
    System.out.println("Negative numbers are not allowed.");
}
}
scanner.close();

for (int i = 0; i < index; i++) {
    System.out.print(numbers[i] + " ");
}
}
```

# PROBLEM-5: OUTPUT



The screenshot displays an IDE interface with the following components:

- EXPLORER:** Shows a project named 'SESSIONAL-2(OOTs)' containing files like `com`, `Employee.class`, `NegativeNumberNotAllowedException.class`, and several `Problem` files.
- Problem5.java:** The main code file is open, showing the following code:

```
1 import java.util.Scanner;
2
3 class NegativeNumberNotAllowedException extends Exception {
4 }
5
6 public class Problem5 {
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9
10        int n = 0;
11        System.out.print("Enter the size of the array: ");
12        n = scanner.nextInt();
13
14        int[] numbers = new int[n];
```
- TERMINAL:** Shows the execution of the program. The user enters 5 for the array size, followed by five numbers: 32, 45, 33, 22, and -2. The program correctly identifies -2 as a negative number and prompts for another input, -5, which is also identified as negative. Finally, it prints the array: `32 45 33 22 5`.