Assignment Day-1&2

Core Java with DS and Algorithms

Name: Joshnitha Rangolu

Task 1: Data Types/Variables

Write a program that declares two integer variables, swaps their values without using a third variable, and prints the result.

```
package day_1and2;

public class SwapWithoutTemp {

public static void main(String [] args) {

int x = 2;

int y = 4;

x=x^y;

y=x^y;

x=x^y;

System.out.println("After swapping");

System.out.println("x="+x);

System.out.println("y="+y);

}
```

FindSecondLargest2.java FindSecondLargest.java Hackerrank3.java

```java
package day_1and2;

public class SwapWithoutTemp {

    public static void main(String [] args) {
        int x = 2;
        int y = 4;

        x=x^y;
        y=x^y;
        x=x^y;

       System.out.println("After swapping");
       System.out.println("x="+x);
       System.out.println("y="+y);
    }

}
```

Console

```
<terminated> SwapWithoutTemp [Java Application] C:\Users\DELL\.p2\pool\plugins\or
After swapping
x=4
y=2
```

Task 2: Operators

Create a program that simulates a simple calculator using command-line arguments to perform and print the result of addition, subtraction, multiplication, and division..

```java
package day_1and2;

public class Calculator {

public static void main(String[] args) {

if (args.length != 3) {

System.out.println("Usage: java Calculator <number1> <operator> <number2>");

return;

}

double num1;

double num2;
```

```
char operator;
try {
num1 = Double.parseDouble(args[0]);
num2 = Double.parseDouble(args[1]);
operator = args[2].charAt(0);
} catch (NumberFormatException e) {
System.out.println("Invalid number format");
return;
}
double result = calculate(num1, num2, operator);
if (Double.isNaN(result)) {
System.out.println("Invalid operator or division by zero");
} else {
System.out.println(num1 + " " + operator + " " + num2 + " = " + result);
}
}
public static double calculate(double num1, double num2, char operator) {
switch (operator) {
case '+':
return num1 + num2;
case '-':
return num1 - num2;
case '*':
return num1 * num2;
case '/':
if (num2 == 0) {
return Double.NaN;
}
```

```
return num1 / num2;

default:

return Double.NaN;

}

}

}
```

Task 3: Control Flow

Write a Java program that reads an integer and prints whether it is a prime number using a for loop and if statements.

```
package day_1and2;

import java.util.Scanner;

public class IsPrime {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter an integer: ");

int number = scanner.nextInt();

boolean isPrime = true;

if (number <= 1) {

isPrime = false;

} else {

for (int i = 2; i <= Math.sqrt(number); i++) {

if (number % i == 0) {

isPrime = false;

break;

}

}

}
```

```
System.out.println(number + (isPrime ? " is a prime number" : " is not a
prime number"));

scanner.close();

}

}
```

FindSecondLargest2.java | FindSecondLargest.java | Hackerrank3.java | SwapWithoutTemp.java | Calculator.java | *IsPrime.java

```
12
13          boolean isPrime = true;
14
15          if (number <= 1) {
16            isPrime = false;
17          } else {
18            for (int i = 2; i <= Math.sqrt(number); i++) {
19              if (number % i == 0) {
20                isPrime = false;
21                break;
22              }
23            }
24          }
25
26          System.out.println(number + (isPrime ? " is a prime number" : " is not a prime number"));
27
28          scanner.close();
29        }
30
31  }
32
```

Console

```
<terminated> IsPrime [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre
Enter an integer: 6
6 is not a prime number
```

Task 4: Constructors

Implement a Matrix class that has a constructor which initializes the dimensions of a matrix and a method to fill the matrix with values

Matrix class:

```
package day_1and2;

public class Matrix {

private int rows;

private int cols;

private int[][] data;

public Matrix(int rows, int cols) {

this.rows = rows;

this.cols = cols;
```

```
this.data = new int[rows][cols];
}
public void fill(int value) {
for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++) {
data[i][j] = value;
}
}
}
public void printMatrix() {
for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++) {
System.out.print(data[i][j] + " ");
}
System.out.println();
}
}
}
```

Main class:

```
package day_1and2;
public class Main {
public static void main(String[] args) {
Matrix myMatrix = new Matrix(3, 4);
myMatrix.fill(1);
System.out.println("Created Matrix:");
myMatrix.printMatrix();
}
```

```
}
```

FindSecondLarges... | FindSecondLarges... | Hackerrank3.java | SwapWithoutTemp.... | Calculator.java | IsPrime.java | Matrix.java

```
12          this.data = new int[rows][cols];
13      }
14
15      public void fill(int value) {
16        for (int i = 0; i < rows; i++) {
17          for (int j = 0; j < cols; j++) {
18            data[i][j] = value;
19          }
20        }
21      }
22
23      public void printMatrix() {
24            for (int i = 0; i < rows; i++) {
25              for (int j = 0; j < cols; j++) {
26                System.out.print(data[i][j] + " ");
27              }
28              System.out.println();
29            }
30          }
31  }
32
```

Console

```
<terminated> Main [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.e
Created Matrix:
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
```

Task 5: Inheritance

Create a Shape class with a method area() and extend it with Circle and Rectangle classes overriding the area() method appropriately.

```
package day_1and2;

public abstract class Shape {

public abstract double area();

}

class Circle extends Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double area() {

return Math.PI * radius * radius;
```

```
}
}
class Rectangle extends Shape {
private double width;
private double height;
public Rectangle(double width, double height) {
this.width = width;
this.height = height;
}
@Override
public double area() {
return width * height;
}
}
public class MainShape {
public static void main(String[] args) {
Shape circle = new Circle(6);
Shape rectangle = new Rectangle(5, 3);
System.out.println("Circle area: " + circle.area());
System.out.println("Rectangle area: " + rectangle.area());
}
}
```

```
Hackerrank3.... SwapWithout... Calculator.java IsPrime.java Matrix.java Main.java Shape.java × MainShape.java
3 public abstract class Shape {
4
5       public abstract double area();
6     }
7
8     class Circle extends Shape {
9
10      private double radius;
11
12      public Circle(double radius) {
13        this.radius = radius;
14      }
15
16      @Override
17      public double area() {
18        return Math.PI * radius * radius;
19      }
20    }
21
22    class Rectangle extends Shape {
23
```

```
Console ×
<terminated> MainShape [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\j
Circle area: 113.09733552923255
Rectangle area: 15.0
```

Task 6: Packages/Classpath

Create a package com.math.operations and include classes for various arithmetic operations. Demonstrate how to compile and run these using the classpath.

Create package:

com.math.operations

Then create arithmetic operations in com.math.operations package.

Add class:

```
package com.math.operation;

public class Add {

public static int add(int a, int b) {

return a + b;

}

}
```

Subtract class:

```
package com.math.operation;

public class Subtract {

public static int subtract(int a, int b) {

return a - b;
```

```
}

}
```

Multiply class:

```
package com.math.operation;

public class Multiply {

public static int multiply(int a, int b) {

return a * b;

}

}
```

Divide class:

```
package com.math.operation;

public class Divide {

public static double divide(int a, int b) {

if (b == 0) {

throw new IllegalArgumentException("Division by zero!");

}

return (double) a / b;

}

}
```

Enter this commands in command prompt:

cd src

javac com/math/operations/Addition.java

javac com/math/operations/Subtraction.java

javac com/math/operations/Multiplication.java

javac com/math/operations/Division.java

TestOperations class:

```java
package com.math.operation;

public class TestOperations {

public static void main(String[] args) {

int a = 10;

int b = 5;

System.out.println("Addition: " + Add.add(a, b));

System.out.println("Subtraction: " + Subtract.subtract(a, b));

System.out.println("Multiplication: " + Multiply.multiply(a, b));

try {

System.out.println("Division: " + Divide.divide(a, b));

} catch (IllegalArgumentException e) {

System.out.println(e.getMessage());

}

}

}
```

- Compile the Main.java file:

javac com/math/operations/Main.java

- Run the Main class using the classpath:

java com.math.operations.Main

```java
package com.math.operation;

public class TestOperations {
    public static void main(String[] args) {
        int a = 10;
        int b = 5;

        System.out.println("Addition: " + Add.add(a, b));
        System.out.println("Subtraction: " + Subtract.subtract(a, b));
        System.out.println("Multiplication: " + Multiply.multiply(a, b));

        try {
            System.out.println("Division: " + Divide.divide(a, b));
        } catch (IllegalArgumentException e) {
            System.out.println(e.getMessage());
        }
    }

}
```

```
Console
<terminated> TestOperations (1) [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
```

Task 7: Basic Exception Handling

Write a program that attempts to divide by zero, catches the ArithmeticException, and provides a custom error message.

```java
package day_1and2;

public class Exception {

public static void main(String[] args) {

int numerator = 10;

int denominator = 0;

try {

double result = divide(numerator, denominator);

System.out.println("Result: " + result);

} catch (ArithmeticException e) {
```

```
System.out.println("Error: Cannot divide by zero.");

}

}

public static double divide(int a, int b) {

return (double) a / b;

}

}
```

FindSecondL... | FindSecondL... | TestOperatio... | Divide.java | Add.java | Subtract.java | Multiply.java | *Exc

```
1  package day_1and2;
2
3  public class Exception {
4      public static void main(String[] args) {
5          int numerator = 10;
6          int denominator = 0;
7
8          try {
9            double result = divide(numerator, denominator);
10           System.out.println("Result: " + result);
11         } catch (ArithmeticException e) {
12           System.out.println("Error: Cannot divide by zero.");
13         }
14       }
15
16       public static double divide(int a, int b) {
17         return (double) a / b;
18       }
19
20 }
21
```

Console ×

<terminated> Exception [Java Application] C:\Users\DELL\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v202

```
Result: Infinity
```