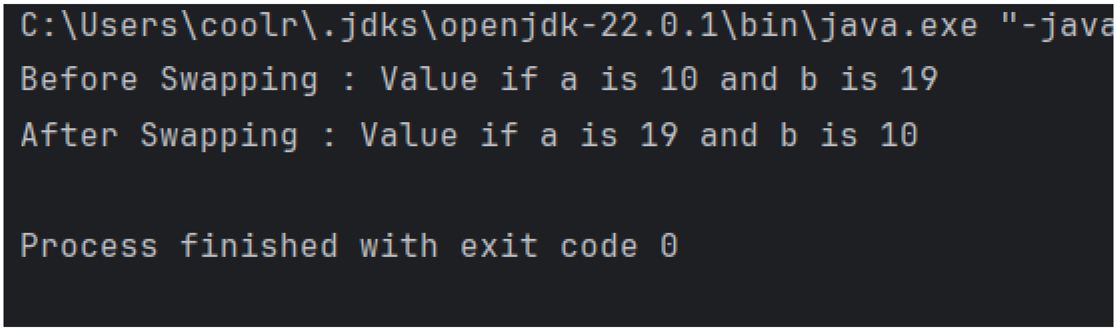
Day 10 Assignment

Part 1

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

```
public class Part_1_Assignment_1 {
   /*
       Write a program that declares two integer variables,
       swaps their values without using a third variable, and prints the
result.
    */
   public static void main(String[] args) {
       int a = 10;
       int b = 19;

       System.out.println("Before Swapping : Value if a is "+a+" and b is "+b);
       b += a;
       a = ~(a-b)+1;
       b = b-a;

       System.out.println("After Swapping : Value if a is "+a+" and b is "+b);
   }
}
```

Output

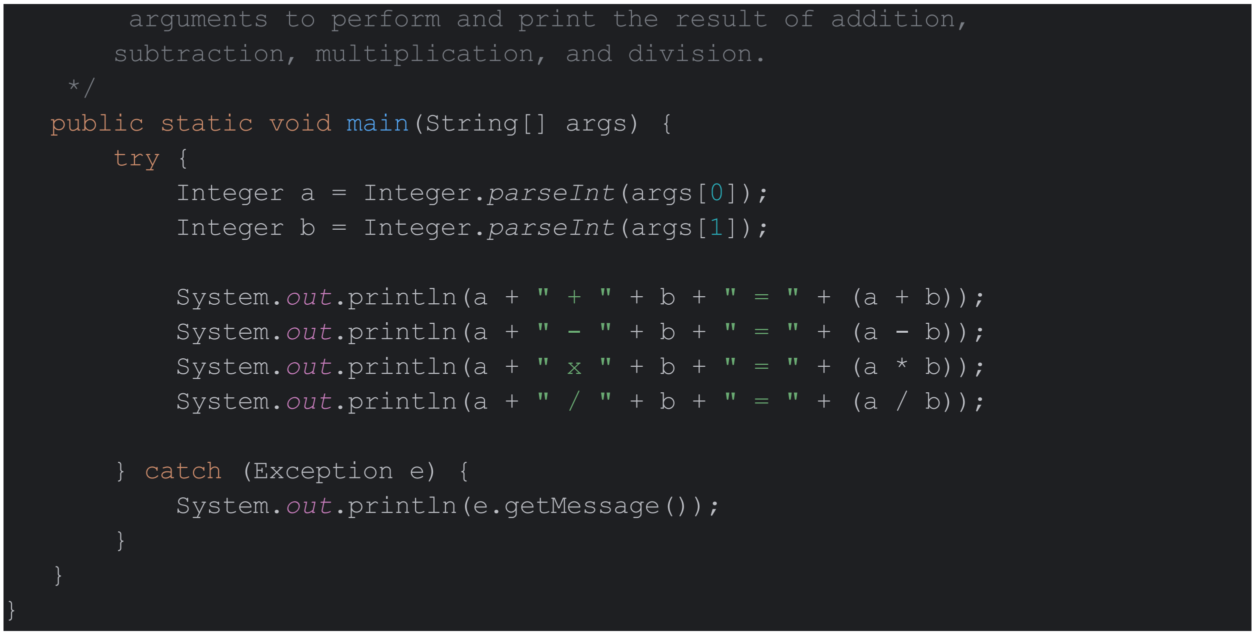


```
C:\Users\coolr\.jdks\openjdk-22.0.1\bin\java.exe "-java
Before Swapping : Value if a is 10 and b is 19
After Swapping : Value if a is 19 and b is 10

Process finished with exit code 0
```

---

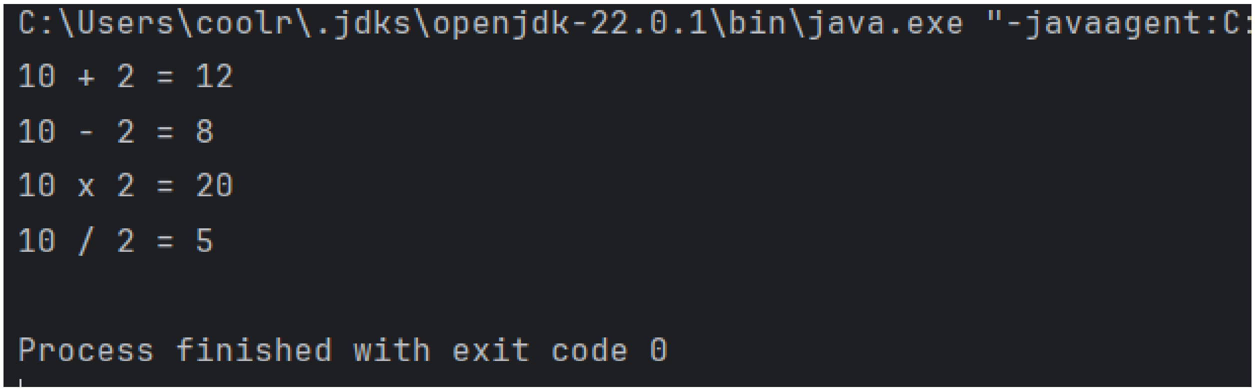
2. **Program that simulates a simple calculator using command-line arguments to perform and print the result of addition, subtraction, multiplication, and division.**

```
public class Part_1_Assignment_2 {
   /*
       Program that simulates a simple calculator using command-line
```



```
         arguments to perform and print the result of addition,
        subtraction, multiplication, and division.
     */
    public static void main(String[] args) {
        try {
            Integer a = Integer.parseInt(args[0]);
            Integer b = Integer.parseInt(args[1]);

            System.out.println(a + " + " + b + " = " + (a + b));
            System.out.println(a + " - " + b + " = " + (a - b));
            System.out.println(a + " x " + b + " = " + (a * b));
            System.out.println(a + " / " + b + " = " + (a / b));

        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

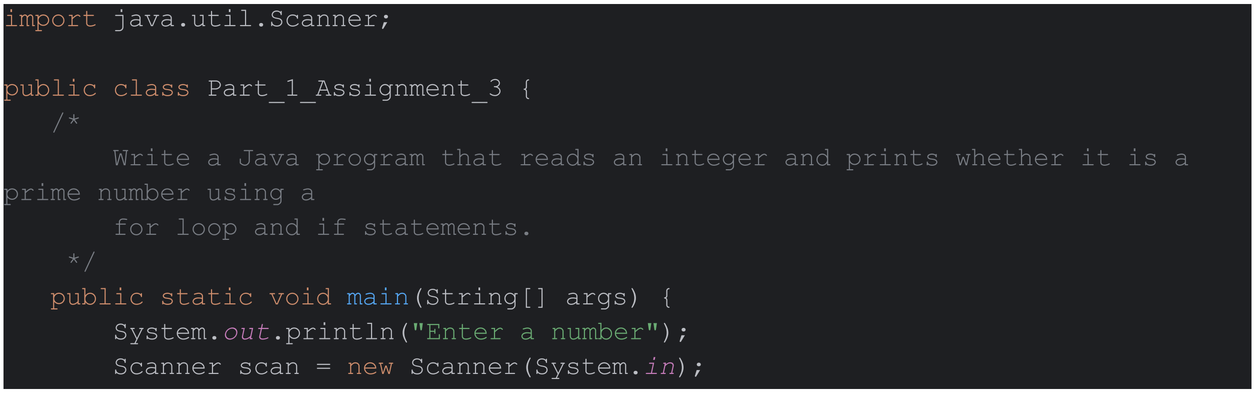
Output



```
C:\Users\coolr\.jdks\openjdk-22.0.1\bin\java.exe "-javaagent:C:
10 + 2 = 12
10 - 2 = 8
10 x 2 = 20
10 / 2 = 5

Process finished with exit code 0
```

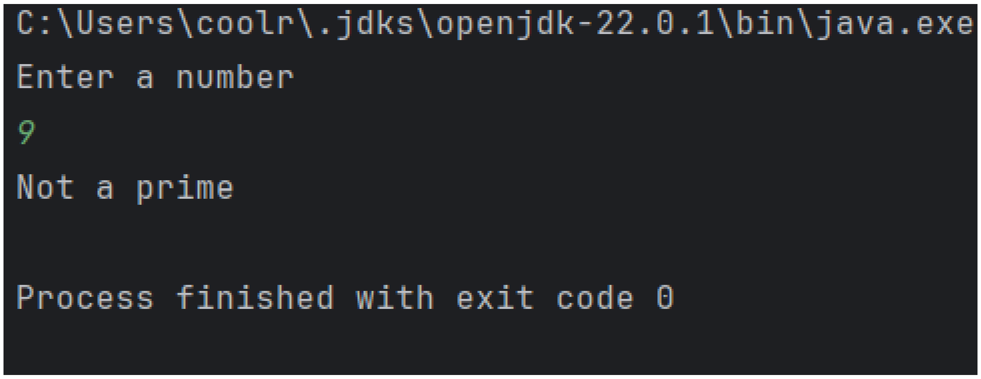
---

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



```
import java.util.Scanner;

public class Part_1_Assignment_3 {
    /*
        Write a Java program that reads an integer and prints whether it is a
prime number using a
        for loop and if statements.
     */
    public static void main(String[] args) {
        System.out.println("Enter a number");
        Scanner scan = new Scanner(System.in);
        int num = scan.nextInt();
        if (num <= 1) {
            System.out.println("Not a prime");
        } else {
            boolean flag = true;
            for (int i = 2; i * i <= num; i++) {
                if (num % i == 0) {
                    flag = false;
                    break;
                }
            }
            if (flag) {
                System.out.println("It is a prime");
            } else {
                System.out.println("Not a prime");
            }
        }

    }
}
```

Output

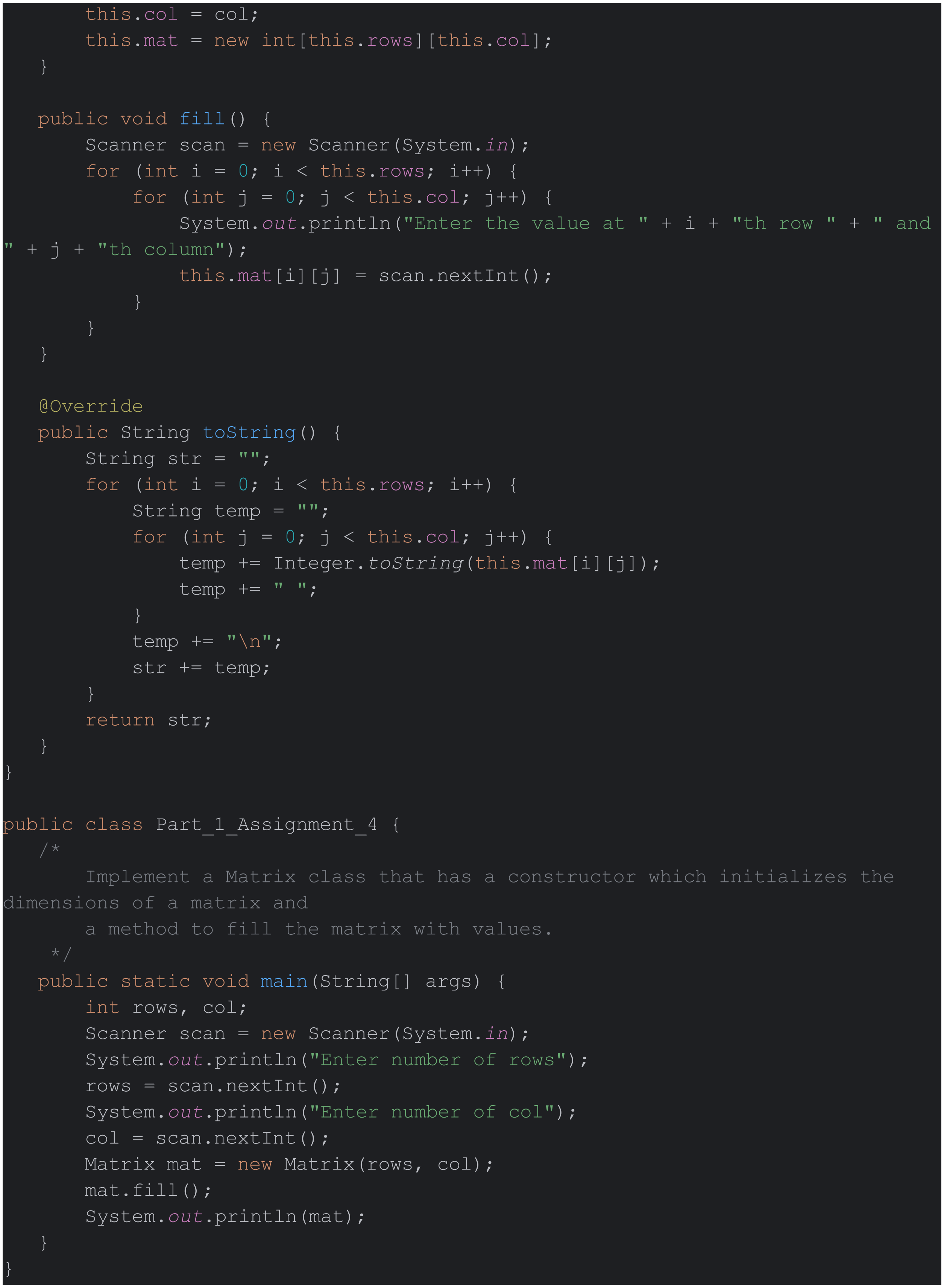


```
C:\Users\coolr\.jdks\openjdk-22.0.1\bin\java.exe
Enter a number
9
Not a prime

Process finished with exit code 0
```

---

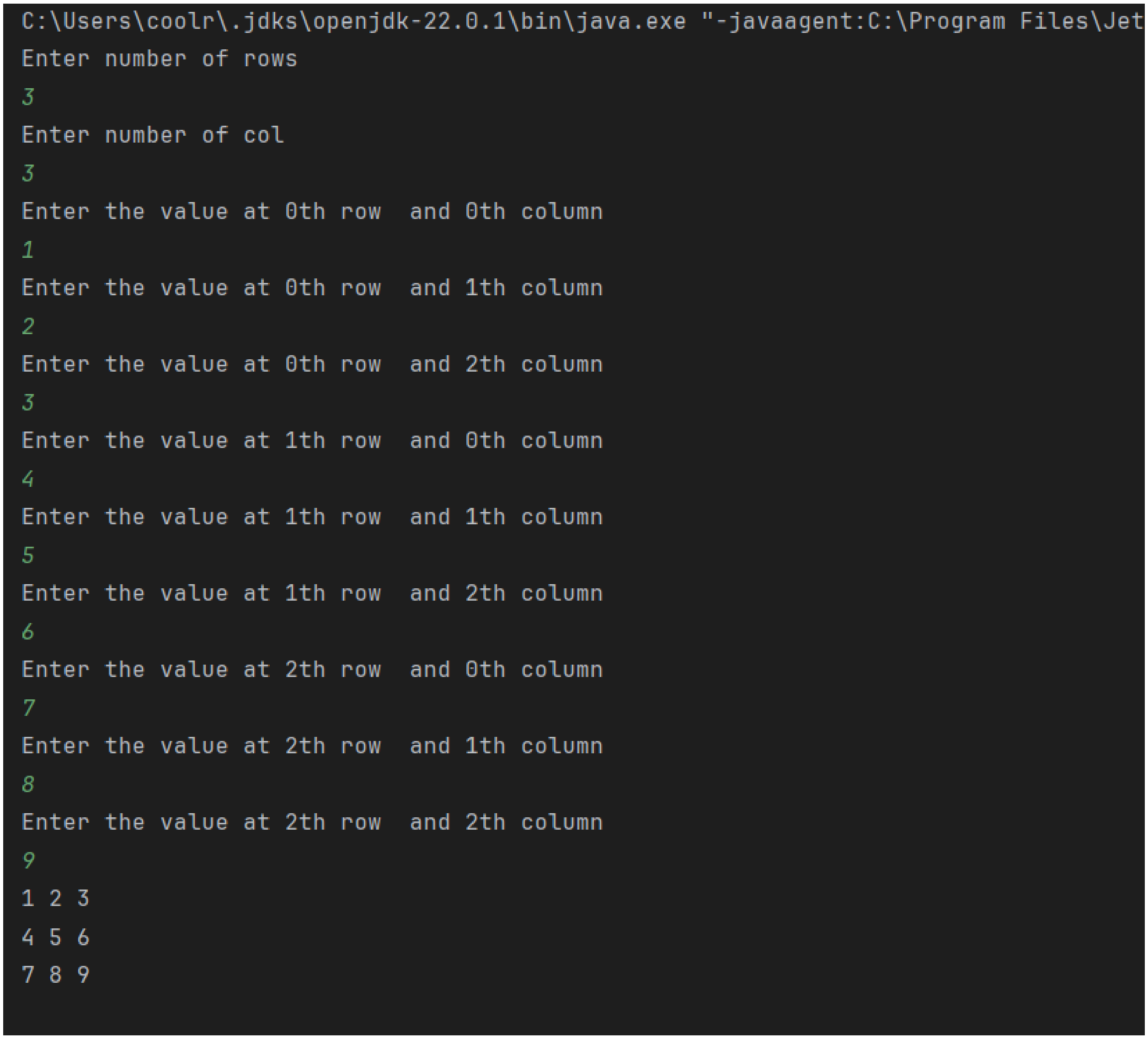
4. **Implement a Matrix class that has a constructor which initializes the dimensions of a matrix and a method to fill the matrix with values.**

```
import java.util.Scanner;

class Matrix {
    private int rows;
    private int col;
    private int[][] mat;

    Matrix(int rows, int col) {
        this.rows = rows;
```



```
       this.col = col;
       this.mat = new int[this.rows][this.col];
   }

   public void fill() {
       Scanner scan = new Scanner(System.in);
       for (int i = 0; i < this.rows; i++) {
           for (int j = 0; j < this.col; j++) {
               System.out.println("Enter the value at " + i + "th row " + " and
" + j + "th column");
               this.mat[i][j] = scan.nextInt();
           }
       }
   }

   @Override
   public String toString() {
       String str = "";
       for (int i = 0; i < this.rows; i++) {
           String temp = "";
           for (int j = 0; j < this.col; j++) {
               temp += Integer.toString(this.mat[i][j]);
               temp += " ";
           }
           temp += "\n";
           str += temp;
       }
       return str;
   }
}

public class Part_1_Assignment_4 {
   /*
       Implement a Matrix class that has a constructor which initializes the
dimensions of a matrix and
       a method to fill the matrix with values.
    */
   public static void main(String[] args) {
       int rows, col;
       Scanner scan = new Scanner(System.in);
       System.out.println("Enter number of rows");
       rows = scan.nextInt();
       System.out.println("Enter number of col");
       col = scan.nextInt();
       Matrix mat = new Matrix(rows, col);
       mat.fill();
       System.out.println(mat);
   }
}
```

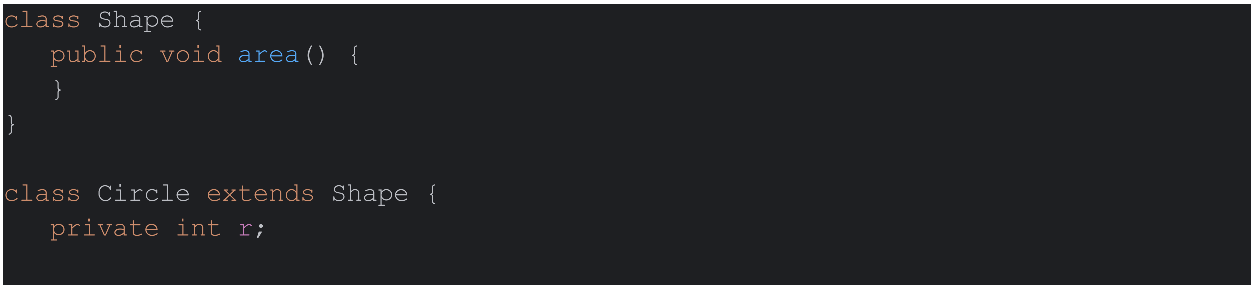
Output



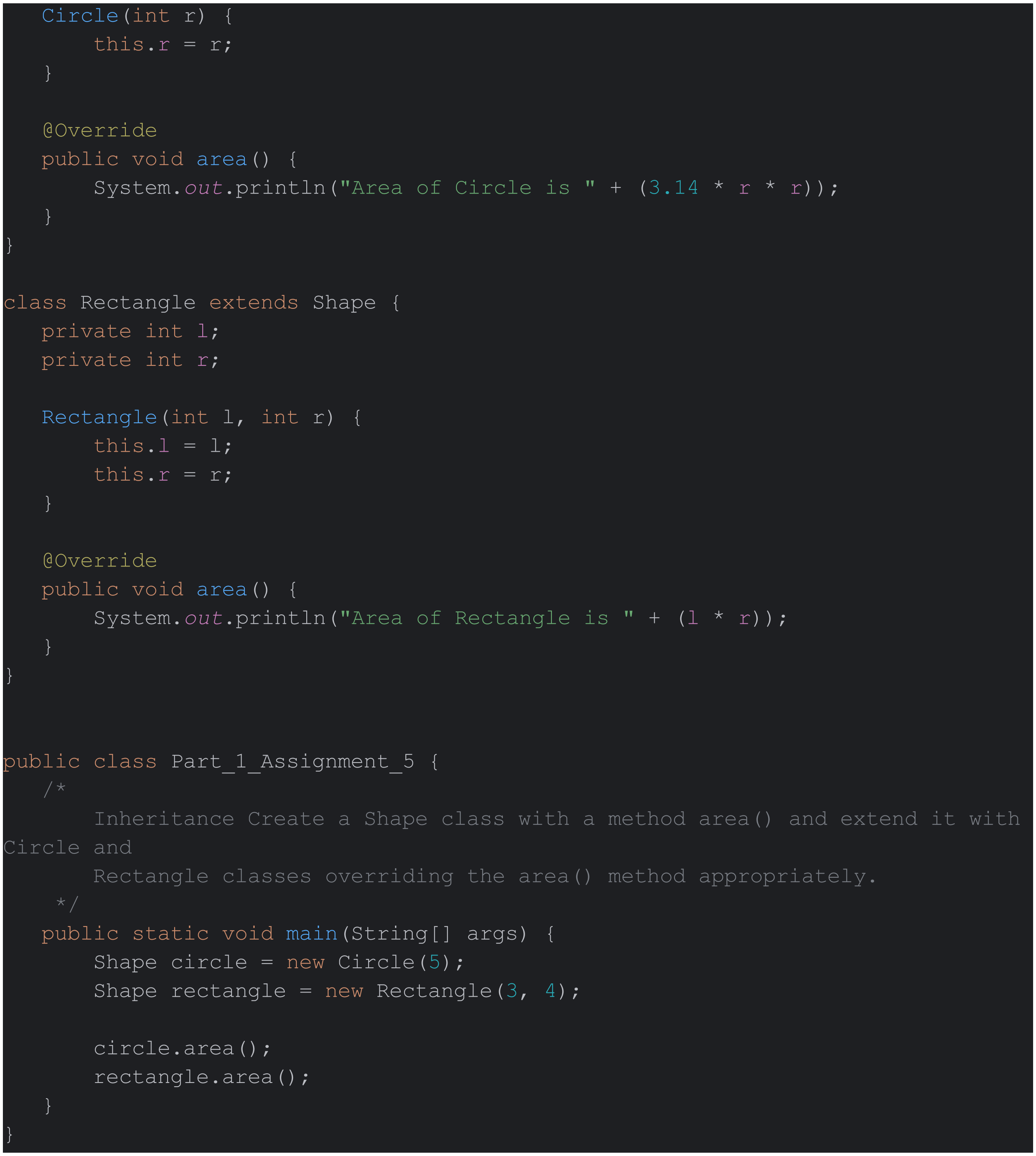
```
C:\Users\coolr\.jdks\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\Jet
Enter number of rows
3
Enter number of col
3
Enter the value at 0th row  and 0th column
1
Enter the value at 0th row  and 1th column
2
Enter the value at 0th row  and 2th column
3
Enter the value at 1th row  and 0th column
4
Enter the value at 1th row  and 1th column
5
Enter the value at 1th row  and 2th column
6
Enter the value at 2th row  and 0th column
7
Enter the value at 2th row  and 1th column
8
Enter the value at 2th row  and 2th column
9
1 2 3
4 5 6
7 8 9
```

---

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

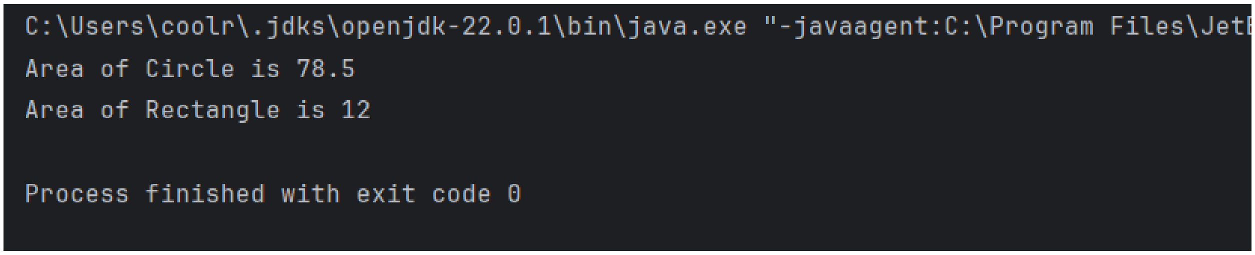


```
class Shape {
   public void area() {
   }
}

class Circle extends Shape {
   private int r;
```



```
    Circle(int r) {
        this.r = r;
    }

    @Override
    public void area() {
        System.out.println("Area of Circle is " + (3.14 * r * r));
    }
}

class Rectangle extends Shape {
    private int l;
    private int r;

    Rectangle(int l, int r) {
        this.l = l;
        this.r = r;
    }

    @Override
    public void area() {
        System.out.println("Area of Rectangle is " + (l * r));
    }
}


public class Part_1_Assignment_5 {
    /*
        Inheritance Create a Shape class with a method area() and extend it with
Circle and
        Rectangle classes overriding the area() method appropriately.
     */
    public static void main(String[] args) {
        Shape circle = new Circle(5);
        Shape rectangle = new Rectangle(3, 4);

        circle.area();
        rectangle.area();
    }
}
```

Output



```
C:\Users\coolr\.jdks\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\Jet
Area of Circle is 78.5
Area of Rectangle is 12

Process finished with exit code 0
```

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

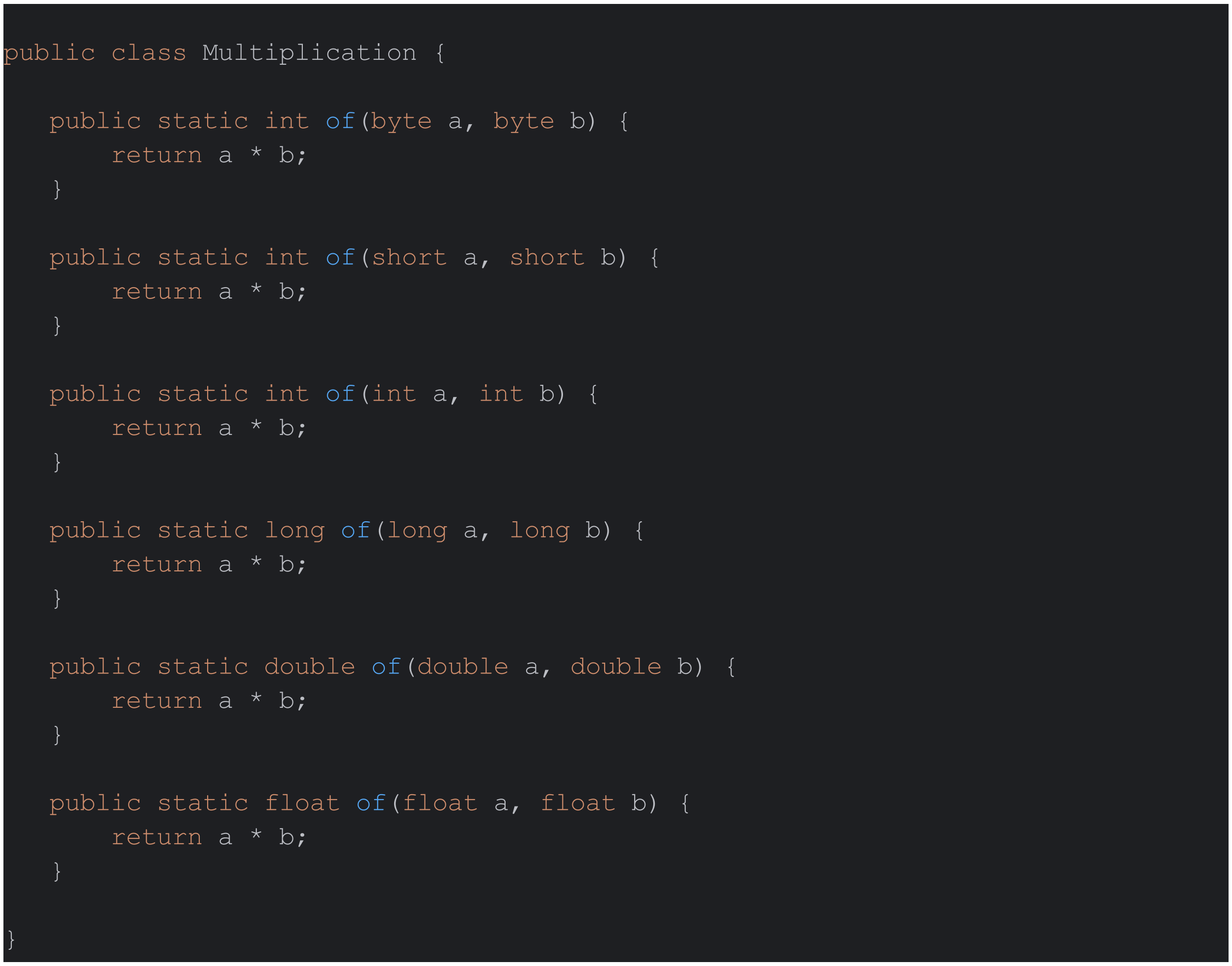
Inside com.math.package

```java
package com.math.operations;

public class Addition {

    public static int of(byte a, byte b) {
        return a + b;
    }

    public static int of(short a, short b) {
        return a + b;
    }

    public static int of(int a, int b) {
        return a + b;
    }

    public static long of(long a, long b) {
        return a + b;
    }

    public static double of(double a, double b) {
        return a + b;
    }

    public static float of(float a, float b) {
        return a + b;
    }

}
```

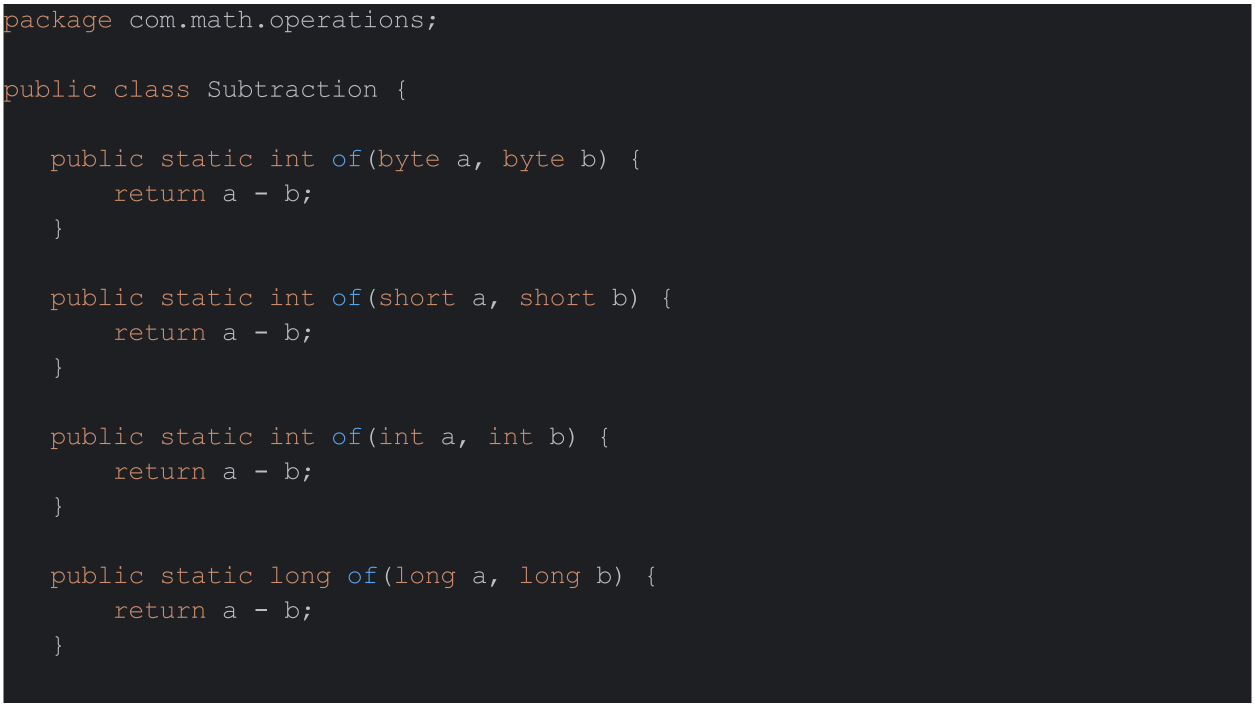
```java
package com.math.operations;

public class Division {

    public static int of(byte a, byte b) {
        return a / b;
    }

    public static int of(short a, short b) {
        return a / b;
    }
    public static int of(int a, int b) {
        return a / b;
    }

    public static long of(long a, long b) {
        return a / b;
    }

    public static double of(double a, double b) {
        return a / b;
    }

    public static float of(float a, float b) {
        return a / b;
    }
}
```

```
package com.math.operations;

public class Modulus {

    public static int of(byte a, byte b) {
        return a % b;
    }

    public static int of(short a, short b) {
        return a % b;
    }

    public static int of(int a, int b) {
        return a % b;
    }

    public static long of(long a, long b) {
        return a % b;
    }

    public static double of(double a, double b) {
        return a % b;
    }

    public static float of(float a, float b) {
        return a % b;
    }

}
```

```
package com.math.operations;
```



```java
public class Multiplication {

    public static int of(byte a, byte b) {
        return a * b;
    }

    public static int of(short a, short b) {
        return a * b;
    }

    public static int of(int a, int b) {
        return a * b;
    }

    public static long of(long a, long b) {
        return a * b;
    }

    public static double of(double a, double b) {
        return a * b;
    }

    public static float of(float a, float b) {
        return a * b;
    }

}
```

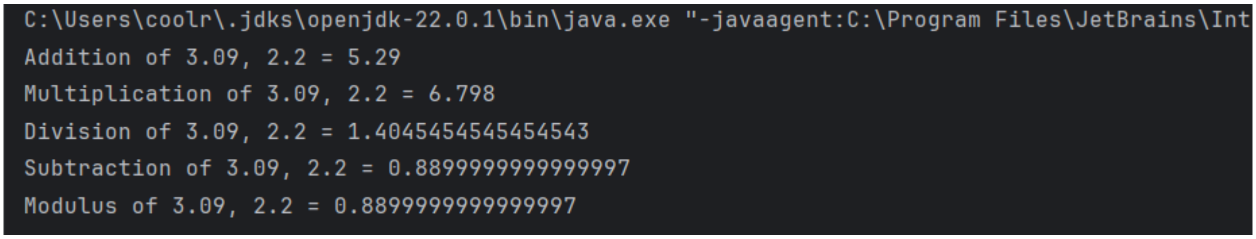


```java
package com.math.operations;

public class Subtraction {

    public static int of(byte a, byte b) {
        return a - b;
    }

    public static int of(short a, short b) {
        return a - b;
    }

    public static int of(int a, int b) {
        return a - b;
    }

    public static long of(long a, long b) {
        return a - b;
    }
    public static double of(double a, double b) {
        return a - b;
    }

    public static float of(float a, float b) {
        return a - b;
    }

}
```

Main Class

```
import com.math.operations.*;

/*
    Create a package com.math.operations and include classes for various arithmetic operations.
    Demonstrate how to compile and run these using the classpath.
*/
public class Part_1_Assignment_6 {
    public static void main(String[] args) {
        System.out.println("Addition of 3.09, 2.2 = " + Addition.of(3.09, 2.2));
        System.out.println("Multiplication of 3.09, 2.2 = " + Multiplication.of(3.09, 2.2));
        System.out.println("Division of 3.09, 2.2 = " + Division.of(3.09, 2.2));
        System.out.println("Subtraction of 3.09, 2.2 = " + Subtraction.of(3.09, 2.2));
        System.out.println("Modulus of 3.09, 2.2 = " + Modulus.of(3.09, 2.2));
    }
}
```

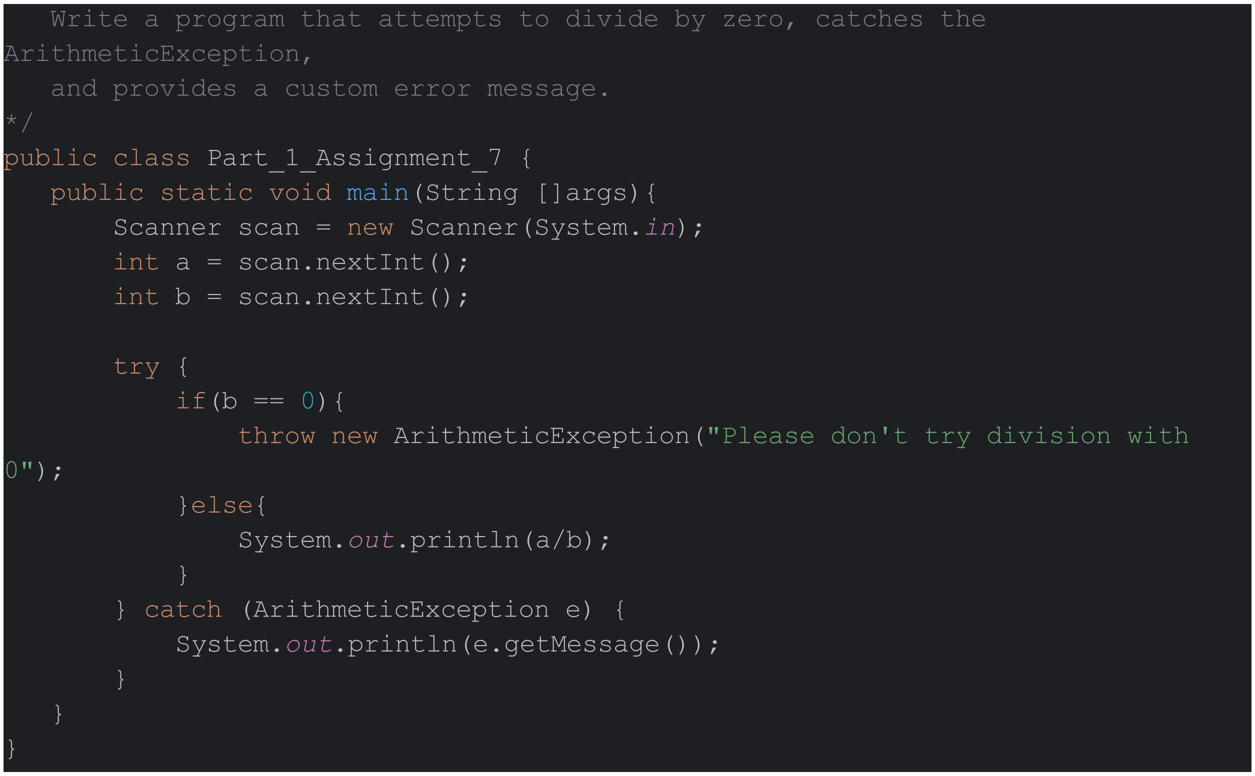
Output



```
C:\Users\coolr\.jdks\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\Int
Addition of 3.09, 2.2 = 5.29
Multiplication of 3.09, 2.2 = 6.798
Division of 3.09, 2.2 = 1.4045454545454543
Subtraction of 3.09, 2.2 = 0.8899999999999997
Modulus of 3.09, 2.2 = 0.8899999999999997
```

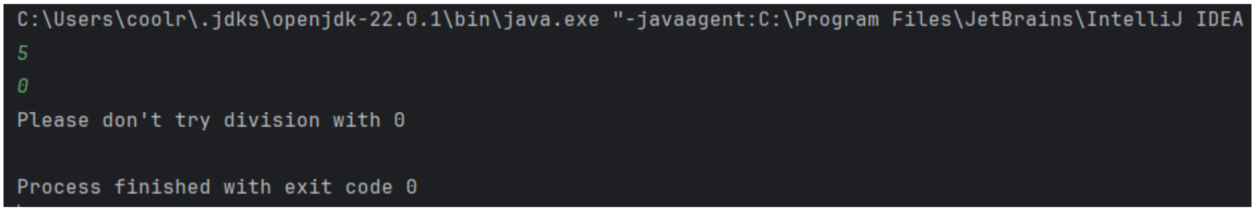
7. **Write a program that attempts to divide by zero, catches the ArithmeticException, and provides a custom error message.**

```
import java.util.Scanner;

/*
```



```
   Write a program that attempts to divide by zero, catches the
ArithmeticException,
   and provides a custom error message.
*/
public class Part_1_Assignment_7 {
   public static void main(String []args){
       Scanner scan = new Scanner(System.in);
       int a = scan.nextInt();
       int b = scan.nextInt();

       try {
           if(b == 0){
               throw new ArithmeticException("Please don't try division with
0");
           }else{
               System.out.println(a/b);
           }
       } catch (ArithmeticException e) {
           System.out.println(e.getMessage());
       }
   }
}
```

Output



```
C:\Users\coolr\.jdks\openjdk-22.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
5
0
Please don't try division with 0

Process finished with exit code 0
```

Tools Used:
IntelliJ IDE
java version "1.8.0_411"
Java(TM) SE Runtime Environment (build 1.8.0_411-b09)
Java HotSpot(TM) Client VM (build 25.411-b09, mixed mode, sharing)