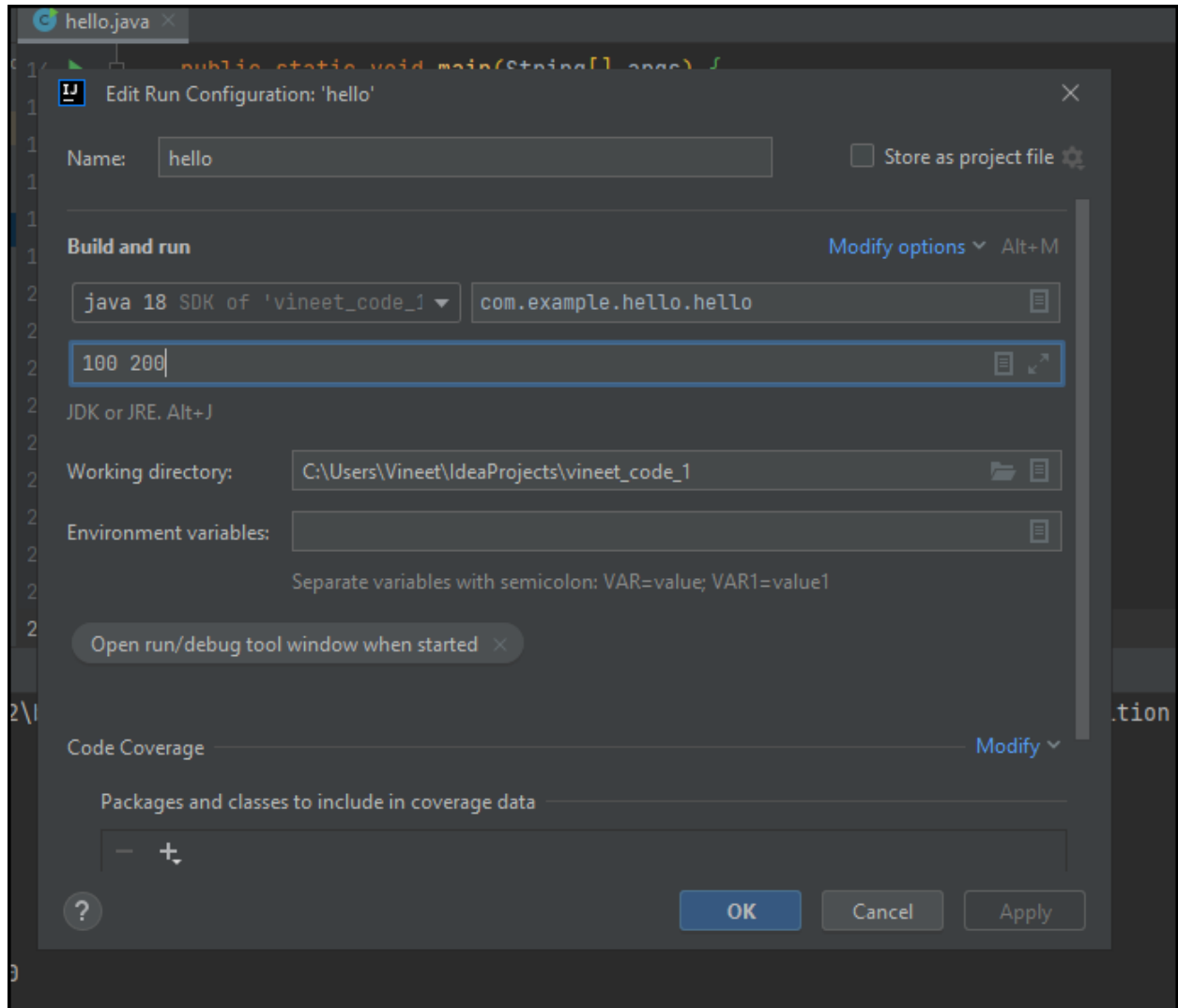
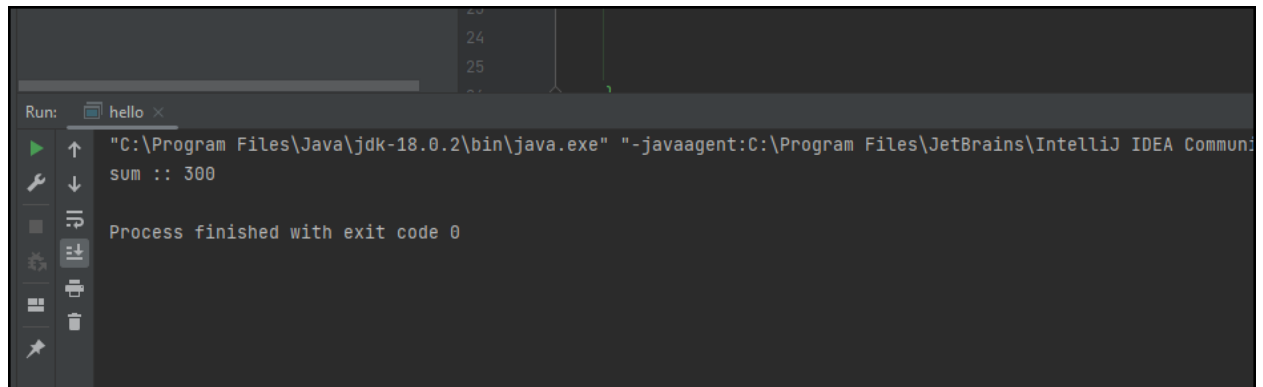


Lab Exercises

1. Write a program to add two input numbers and display the result by printing it out on the console. Use a main method and pass arguments from the command line prompt. (Java Language Constructs)





```
Run: hello x
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi:
sum :: 300

Process finished with exit code 0
```

```
package com.example.hello;
import java.util.*;
import java.lang.*;

// sout-- printing single lines
// psvm --> main file

import java.util.Scanner;

public class hello {

    public static void main(String[] args) {

        int num1 = Integer.parseInt(args[0]) ;
        int num2 = Integer.parseInt(args[1]) ;

        int sum = num1 + num2 ;

        System.out.println("sum :: " + sum);

    }
}
```

2. Understand the usage of as many keywords as possible by writing a program to demonstrate the same. (Java Language Constructs)

```
public class hello {  
  
    public static void main(String[] args) {  
  
        int num1 ;  
        float f_num ;  
        double d_num ;  
        char c = 'A' ;  
        boolean bool = true ;  
  
        num1 = 100 ;  
        if(num1 > 200)  
            System.out.println("yes");  
        else  
            System.out.println("no");  
    }  
}
```

3. Write a program/multiple programs to illustrate the usage of the following: if statement, for loop(simple and enhanced), while loop, do-while loop, switch statement. Use simple logic such as addition of numbers. (Control Constructs)

```
package com.example.hello;  
import java.util.*;  
import java.lang.* ;  
  
// sout-- printing single lines  
// psvm --> main file  
  
import java.util.Scanner;
```

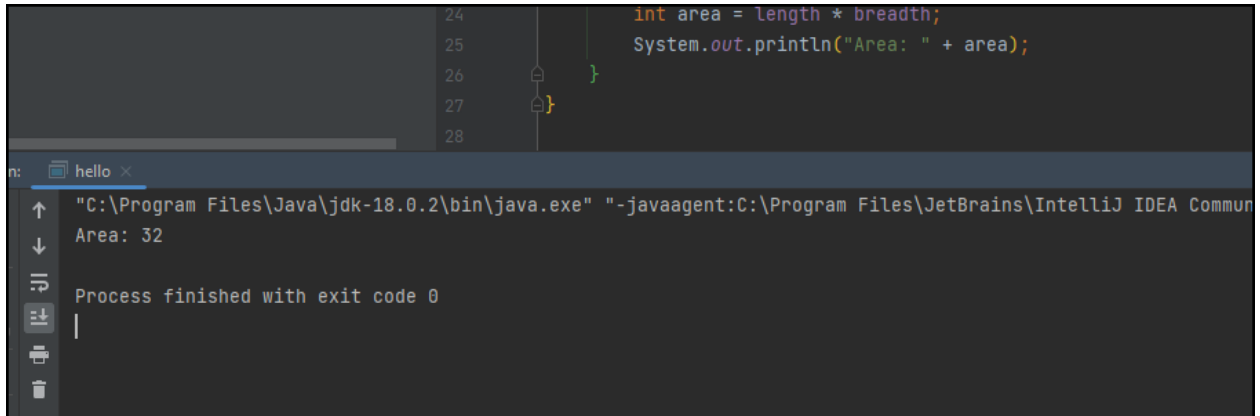
```

public class hello {
    public static void main(String[] args) {
        int num1 = 100 ;
        if(num1 > 200)
            System.out.println("yes");
        else
            System.out.println("no");
        for(int i = 0 ; i < 10 ; i++ )
            System.out.println("hello");
        int [] arr = {10, 20, 30} ;
        for(int i : arr) System.out.println(i);
    }
    int num2 = 3 ;
    while(num2 < 5) {
        System.out.println("hello");
        num2++;
    }
    do {
        System.out.println("hello");
        num2++ ;
    }while(num2 <10)
    int day = 4;
    switch (day) {
        case 1:
            System.out.println("Monday");
            break;
        case 2:
            System.out.println("Tuesday");
            break;
        case 3:
            System.out.println("Wednesday");
            break;
        case 4:
            System.out.println("Thursday");
            break;
        case 5:
            System.out.println("Friday");
            break;
        case 6:
            System.out.println("Saturday");
            break;
        case 7:
            System.out.println("Sunday");
            break;
    }
}

```

4. Write programs to illustrate encapsulation, inheritance, polymorphism, overriding, overloading, interfaces. (OO concepts)

Encapsulation :



```
24         int area = length * breadth;
25         System.out.println("Area: " + area);
26     }
27 }
28
```

hello x

"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Commun

Area: 32

Process finished with exit code 0

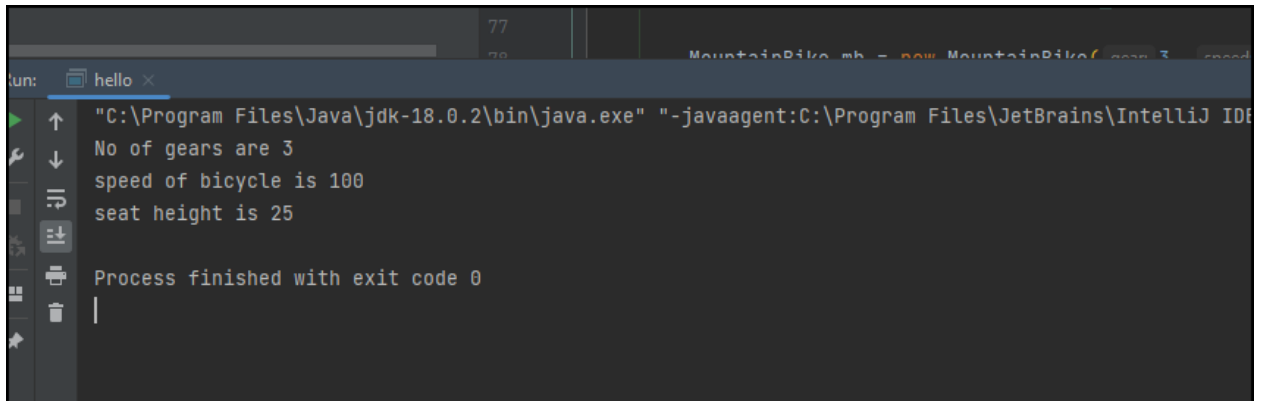
```
class Area {

    int length;
    int breadth;

    // constructor to initialize values
    Area(int length, int breadth) {
        this.length = length;
        this.breadth = breadth;
    }
    // method to calculate area
    public void getArea() {
        int area = length * breadth;
        System.out.println("Area: " + area);
    }
}

public class hello {
    public static void main(String[] args) {
        Area rectangle = new Area(2, 16);
        rectangle.getArea();
    }
}
```

Inheritance :



```
Run: hello x
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDE
No of gears are 3
speed of bicycle is 100
seat height is 25
Process finished with exit code 0
```

```

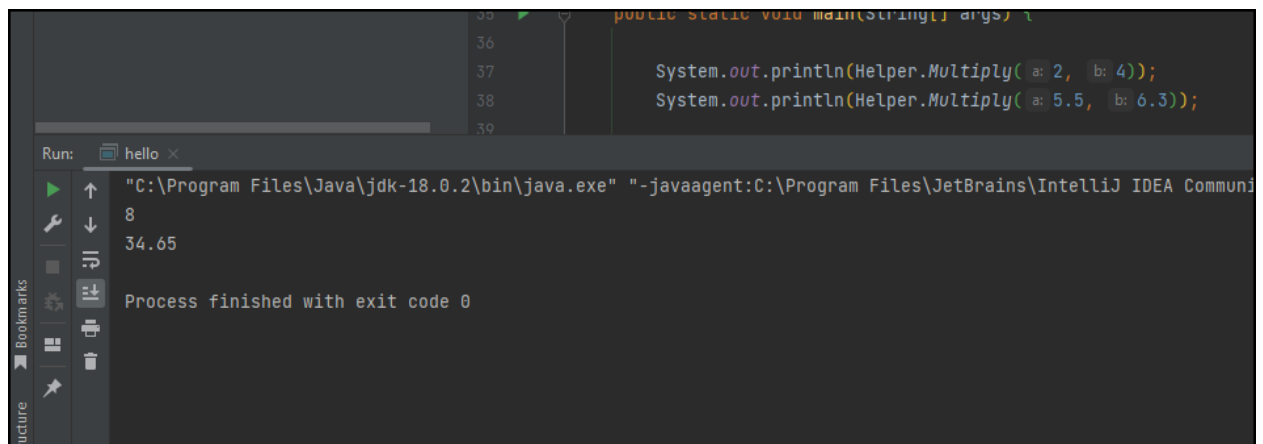
class Bicycle {
    public int gear;
    public int speed;
    public Bicycle(int gear, int speed)
    {
        this.gear = gear;
        this.speed = speed;
    }
    public void applyBrake(int decrement)
    {
        speed -= decrement;
    }
    public void speedUp(int increment)
    {
        speed += increment;
    }
    public String toString()
    {
        return ("No of gears are " + gear + "\n"
            + "speed of bicycle is " + speed);
    }
}
// derived class
class MountainBike extends Bicycle {
    public int seatHeight;
    public MountainBike(int gear, int speed,
        int startHeight)
    {
        super(gear, speed);
        seatHeight = startHeight;
    }
    public void setHeight(int newValue)
    {
        seatHeight = newValue;
    }
    // overriding toString() method
    // of Bicycle to print more info
    @Override public String toString()
    {
        return (super.toString() + "\nseat height is "
            + seatHeight);
    }
}
public class hello {

    public static void main(String[] args) {
        MountainBike mb = new MountainBike(3, 100, 25);
        System.out.println(mb.toString());
    }
}

```

Polymorphism :

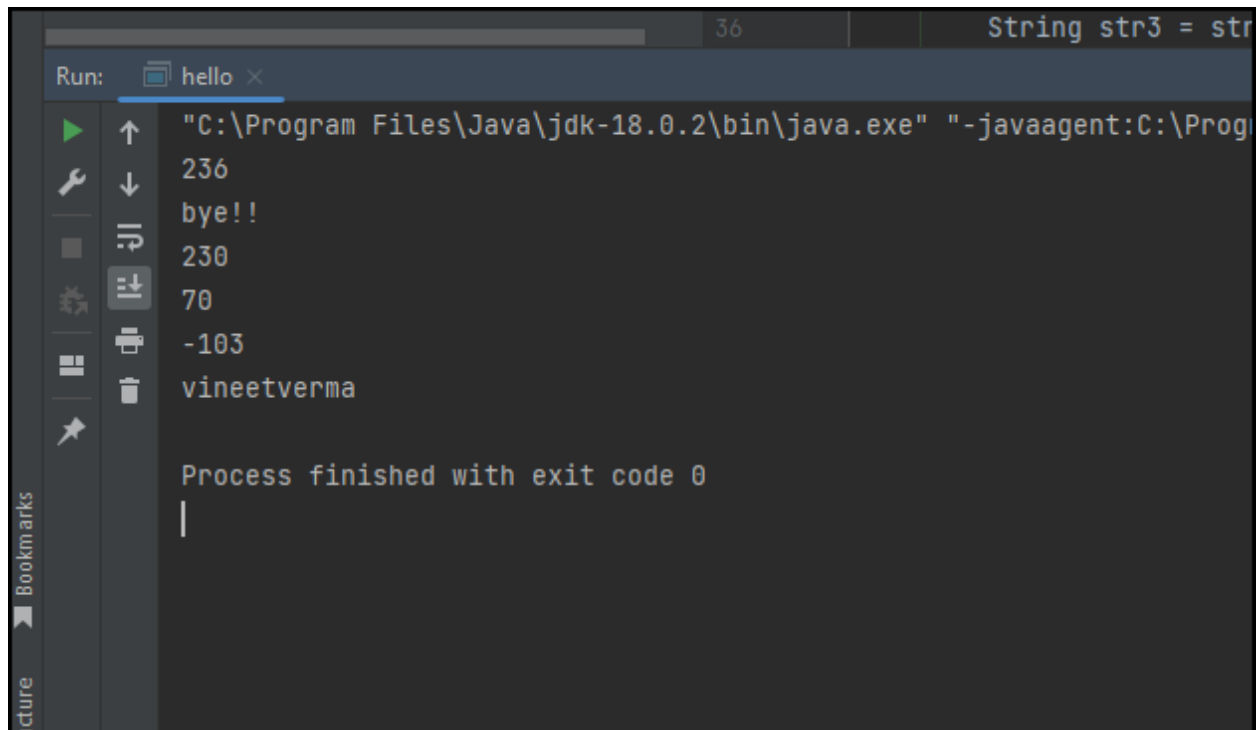
```
class Helper {  
    // Method with 2 integer parameters  
    static int Multiply(int a, int b)  
    {  
        // Returns product of integer numbers  
        return a * b;  
    }  
    // Method 2  
    // With same name but with 2 double parameters  
    static double Multiply(double a, double b)  
    {  
        // Returns product of double numbers  
        return a * b;  
    }  
}  
public class hello {  
    public static void main(String[] args) {  
        System.out.println(Helper.Multiply(2, 4));  
        System.out.println(Helper.Multiply(5.5, 6.3));  
    }  
}
```



The screenshot shows the IntelliJ IDEA IDE with a Java program being executed. The Run window at the bottom displays the output of the program, which is the result of the polymorphic method calls. The output shows the integer result 8 and the double result 34.65, followed by the message "Process finished with exit code 0".

```
Run: hello x  
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi  
8  
34.65  
Process finished with exit code 0
```


5. Write a program/multiple programs to illustrate the usage of the following operators: +, -, *, /, %, =, ==, !=, ++, --, >, <, >=, <=, &, |, ^, ||, &&, !, &=, |=, ^=, ?:, String concatenation operator (+). (Operator basics)



```
Run: hello x
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Progr
236
bye!!
230
70
-103
vineetverma

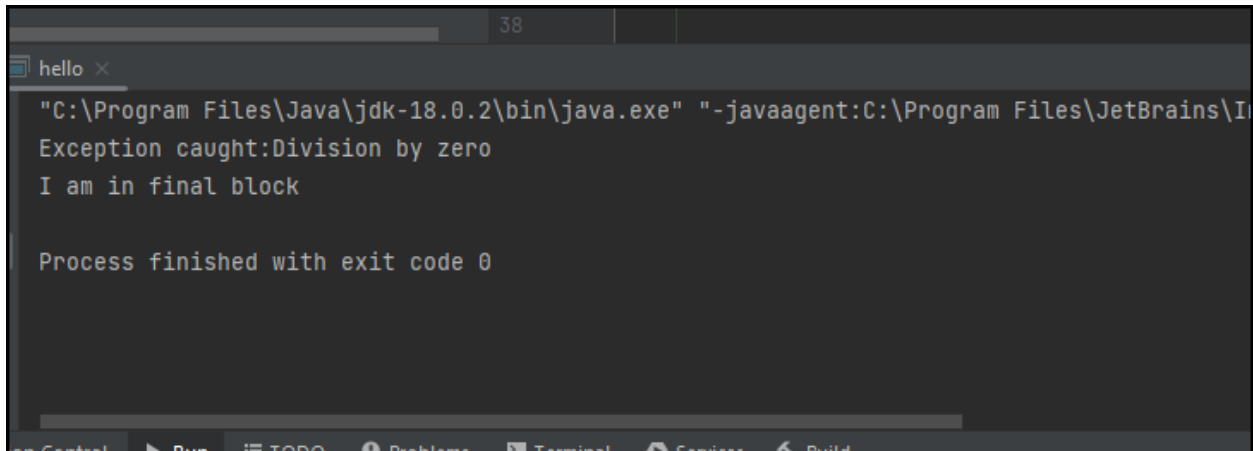
Process finished with exit code 0
```

```
public class hello {

    public static void main(String[] args) {
        int a ,b ;
        a =100 ;
        b =200 ;
        System.out.println(a|b);
        a++ ;
        b-- ; --b ; ++a ;
        if(a==b) System.out.println("hello");
        if(a!=b) System.out.println("bye!!");
        System.out.println((a|b ));
        System.out.println((a&b ));
        System.out.println((~a ));

        String str1 = "vineet" ;
        String str2 = "verma" ;
        String str3 = str1 + str2 ;
        System.out.println(str3);
    }
}
```

6. Write a program which uses a try block, multiple catch blocks and a finally block. Use specific exceptions first followed by generic ones, then interchange the usage to see the difference. (Exception Handling)



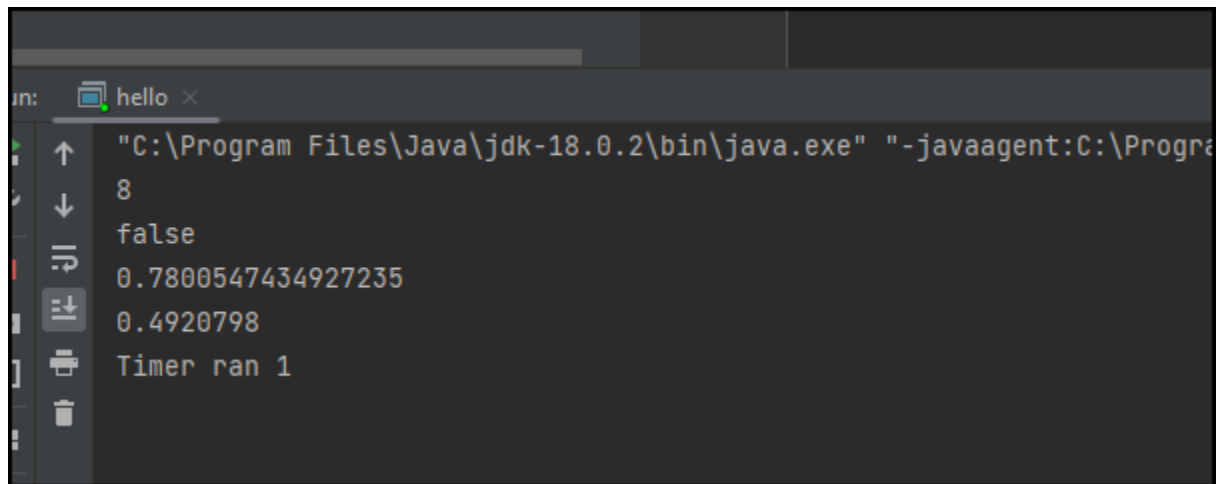
```
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\I
Exception caught:Division by zero
I am in final block

Process finished with exit code 0
```

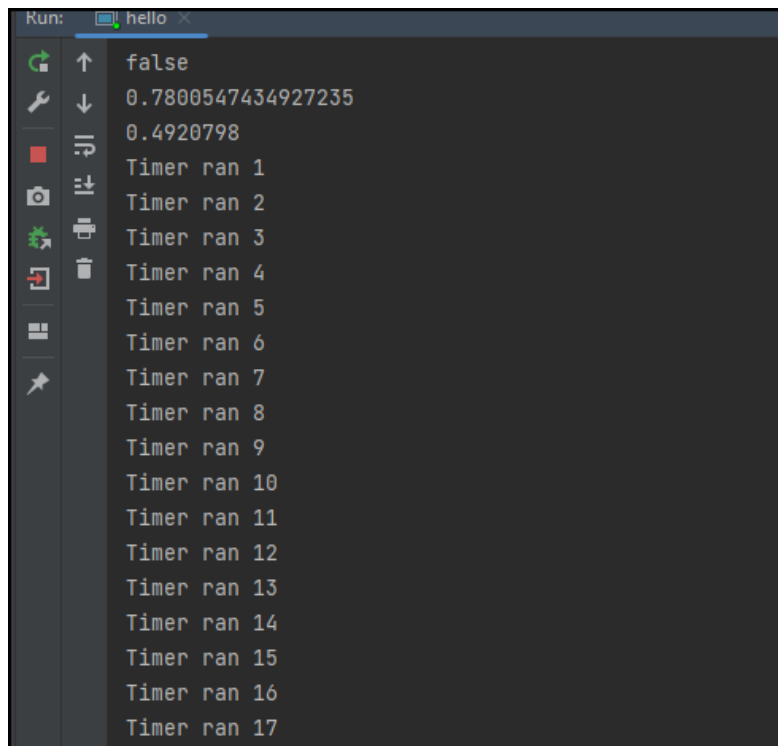
```
public class hello {
    public static void main(String[] args) {
        int a = 10, b = 5, c = 5, result;
        try {
            result = a / (b - c);
            System.out.println("result" + result);
        }
        catch (ArithmeticException e) {
            System.out.println("Exception caught:Division by zero");
        }
        catch (Exception e) {
            System.out.println("You should not divide the number by zero! ");
        }
        finally {
            System.out.println("I am in final block");
        }
    }
}
```

7. Write a program which uses the Java utility classes from the java.util and java.math packages.(Framework Utility classes)

Java util packages :



```
in: hello x
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent:C:\Progra
8
false
0.7800547434927235
0.4920798
Timer ran 1
```



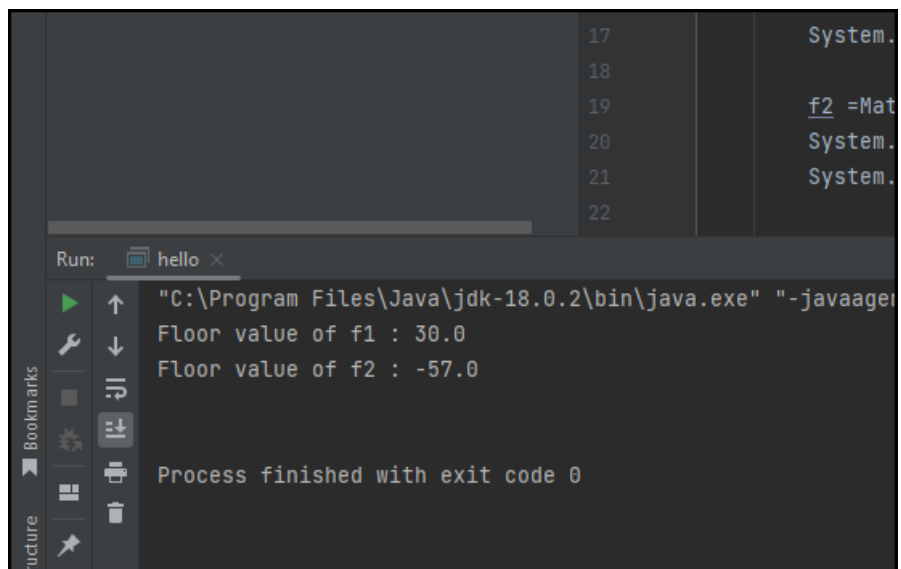
```
Run: hello x
false
0.7800547434927235
0.4920798
Timer ran 1
Timer ran 2
Timer ran 3
Timer ran 4
Timer ran 5
Timer ran 6
Timer ran 7
Timer ran 8
Timer ran 9
Timer ran 10
Timer ran 11
Timer ran 12
Timer ran 13
Timer ran 14
Timer ran 15
Timer ran 16
Timer ran 17
```

```

class Helper extends TimerTask
{
    public static int i = 0;
    public void run()
    {
        System.out.println("Timer ran " + ++i);
    }
}
public class hello {
    public static void main(String[] args) {
        Random random = new Random();
        System.out.println(random.nextInt(10));
        System.out.println(random.nextBoolean());
        System.out.println(random.nextDouble());
        System.out.println(random.nextFloat());
        Timer timer = new Timer();
        TimerTask task = new Helper();
        timer.schedule(task, 2000, 5000);
    }
}

```

Java Math package :



The screenshot shows an IDE with a Java class named 'hello' and its execution output. The code defines a 'Helper' class that extends 'TimerTask' and prints a message every 2000ms for 5000ms. The 'hello' class uses 'Random' to generate various values and schedules the 'Helper' task. The output shows the execution of the program, including the floor values of 'f1' and 'f2'.

```

17      System.out.println("Timer ran " + ++i);
18
19      }
20
21      }
22
Run: hello x
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" "-javaagent
Floor value of f1 : 30.0
Floor value of f2 : -57.0
Process finished with exit code 0

```

```
public class hello {  
  
    public static void main(String[] args) {  
        double f1 = 30.56, f2 = -56.34;  
        f1 =Math.floor(f1);  
        System.out.println("Floor value of f1 : "+f1);  
        f2 =Math.floor(f2);  
        System.out.println("Floor value of f2 : "+f2);  
        System.out.println("");  
    }  
}
```

Vineet Kumar Verma