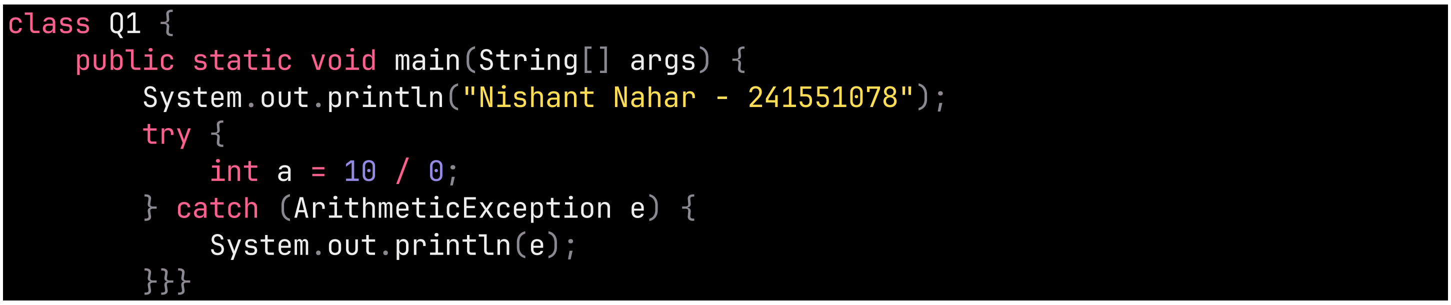
# OOPJ LAB ASSIGNMENT – 7

Name – Nishant Nahar

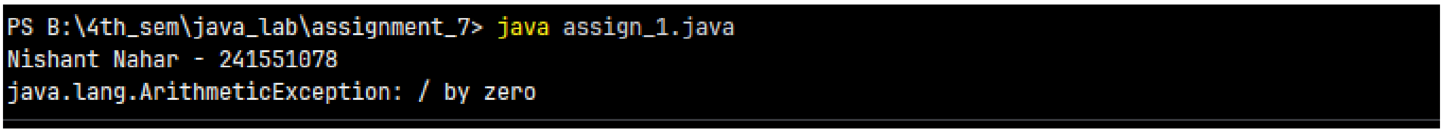
Roll Number – 241551078

1. Write a program to handle ArithmeticException during division of two integers.



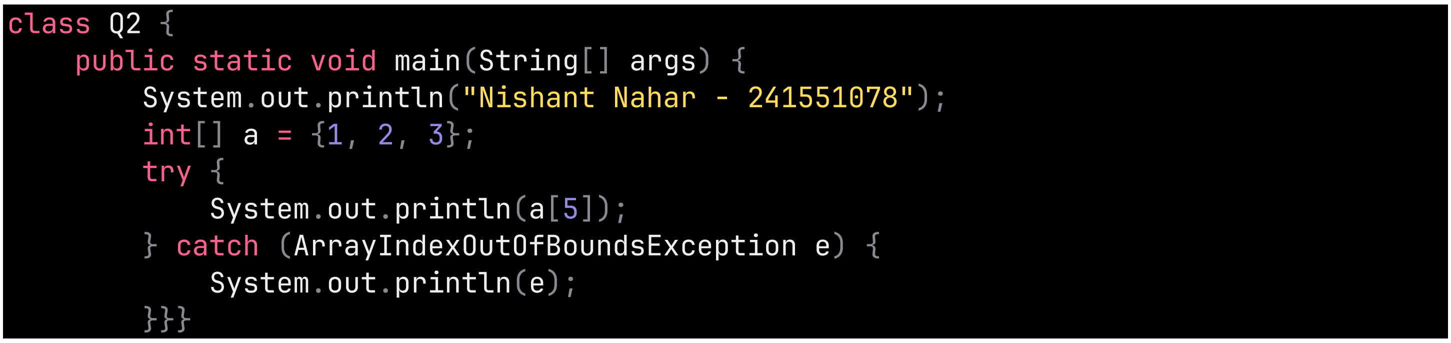
```
class Q1 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
            int a = 10 / 0;
        } catch (ArithmeticException e) {
            System.out.println(e);
        }}}
```

OUTPUT



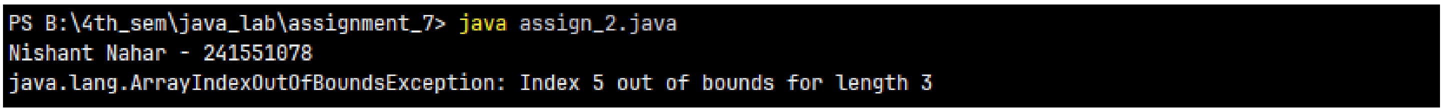
```
PS B:\4th_sem\java_lab\assignment_7> java assign_1.java
Nishant Nahar - 241551078
java.lang.ArithmeticException: / by zero
```

2. Write a program to demonstrate and handle ArrayIndexOutOfBoundsException.



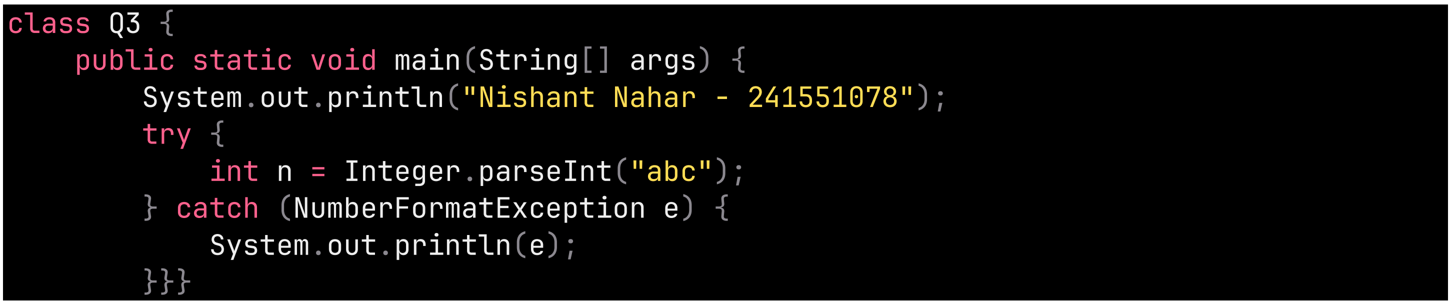
```
class Q2 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        int[] a = {1, 2, 3};
        try {
            System.out.println(a[5]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println(e);
        }}}
```

OUTPUT



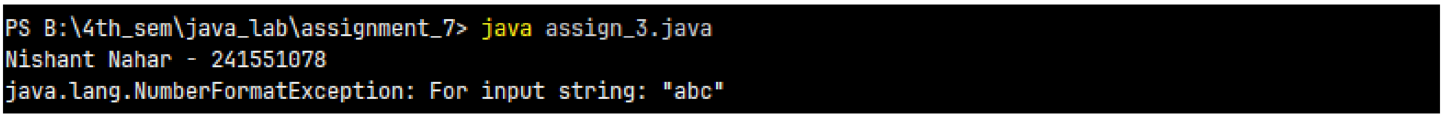
```
PS B:\4th_sem\java_lab\assignment_7> java assign_2.java
Nishant Nahar - 241551078
java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3
```

3. Write a program to handle NumberFormatException when converting a string to an integer.



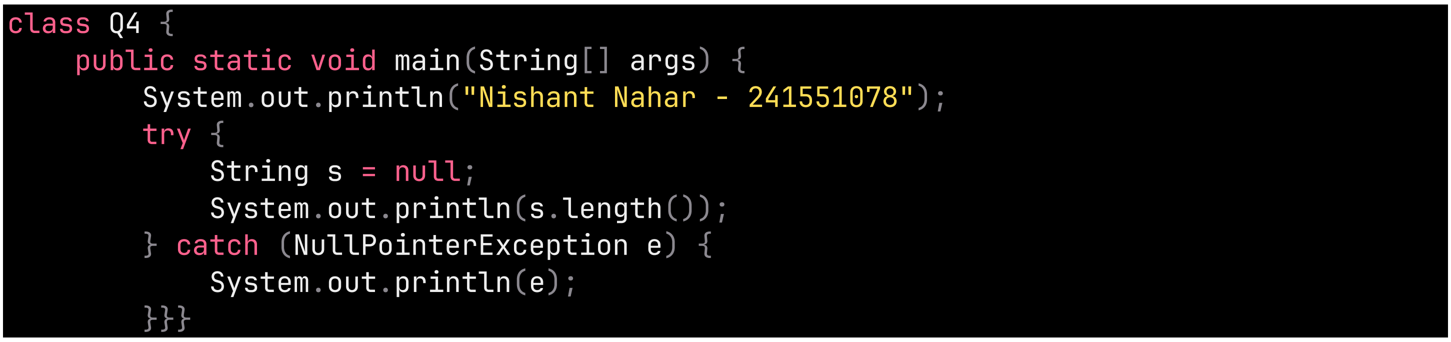
```
class Q3 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
            int n = Integer.parseInt("abc");
        } catch (NumberFormatException e) {
            System.out.println(e);
        }}}
```

OUTPUT



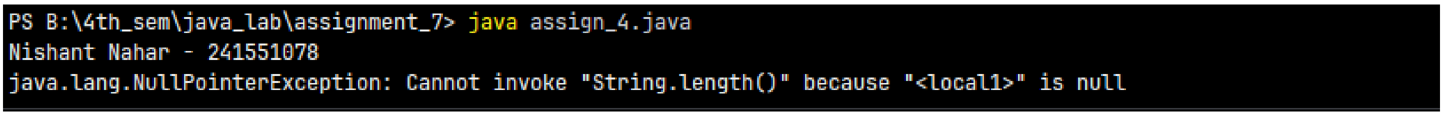
```
PS B:\4th_sem\java_lab\assignment_7> java assign_3.java
Nishant Nahar - 241551078
java.lang.NumberFormatException: For input string: "abc"
```

4. Write a program that safely handles NullPointerException while accessing object methods.



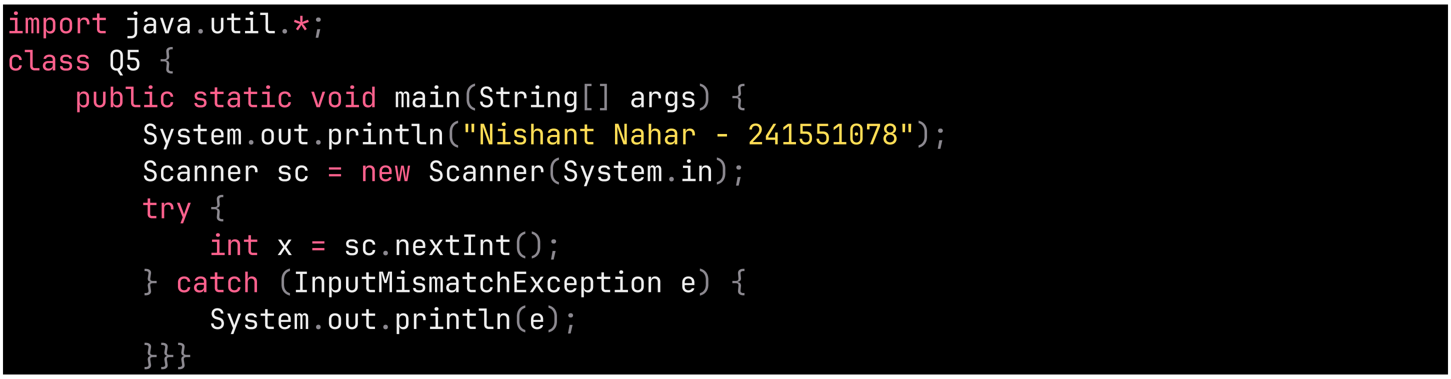
```
class Q4 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
            String s = null;
            System.out.println(s.length());
        } catch (NullPointerException e) {
            System.out.println(e);
        }}}
```

OUTPUT



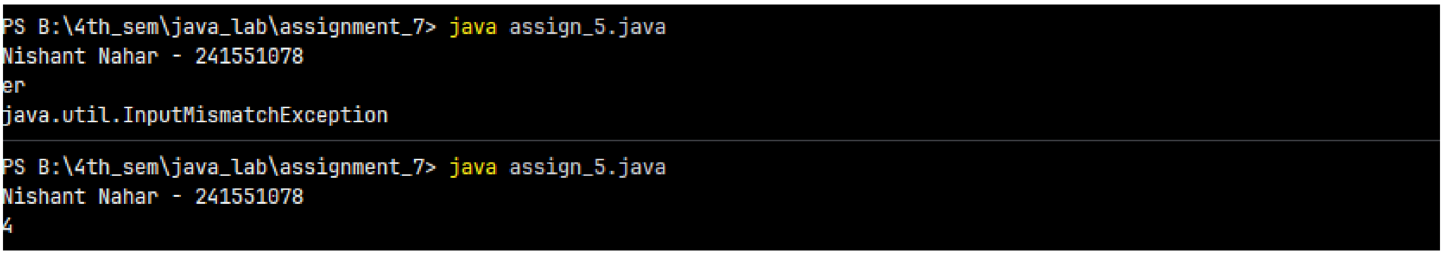
```
PS B:\4th_sem\java_lab\assignment_7> java assign_4.java
Nishant Nahar - 241551078
java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null
```

5. Write a program to handle InputMismatchException using the Scanner class.



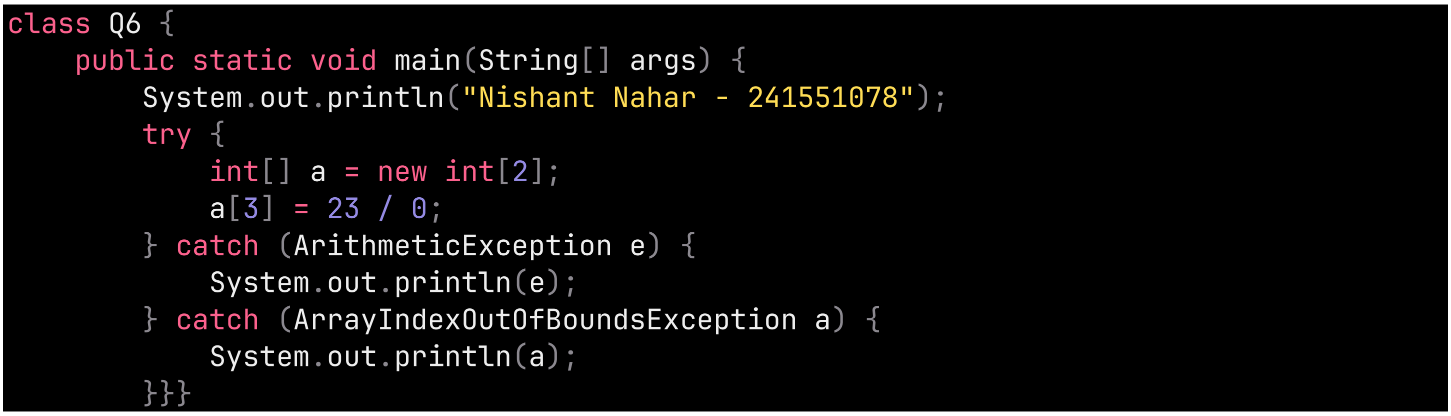
```
import java.util.*;
class Q5 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        Scanner sc = new Scanner(System.in);
        try {
            int x = sc.nextInt();
        } catch (InputMismatchException e) {
            System.out.println(e);
        }}}
```

OUTPUT



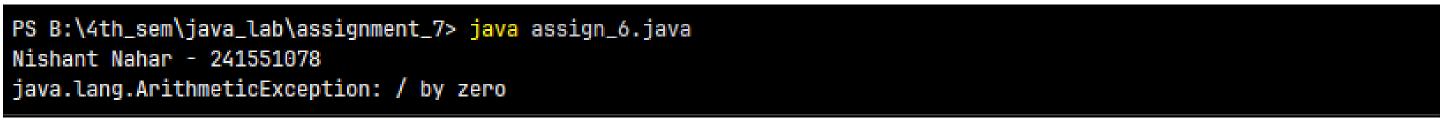
```
PS B:\4th_sem\java_lab\assignment_7> java assign_5.java
Nishant Nahar - 241551078
er
java.util.InputMismatchException

PS B:\4th_sem\java_lab\assignment_7> java assign_5.java
Nishant Nahar - 241551078
4
```

6. Write a program that uses multiple catch blocks for different runtime exceptions.



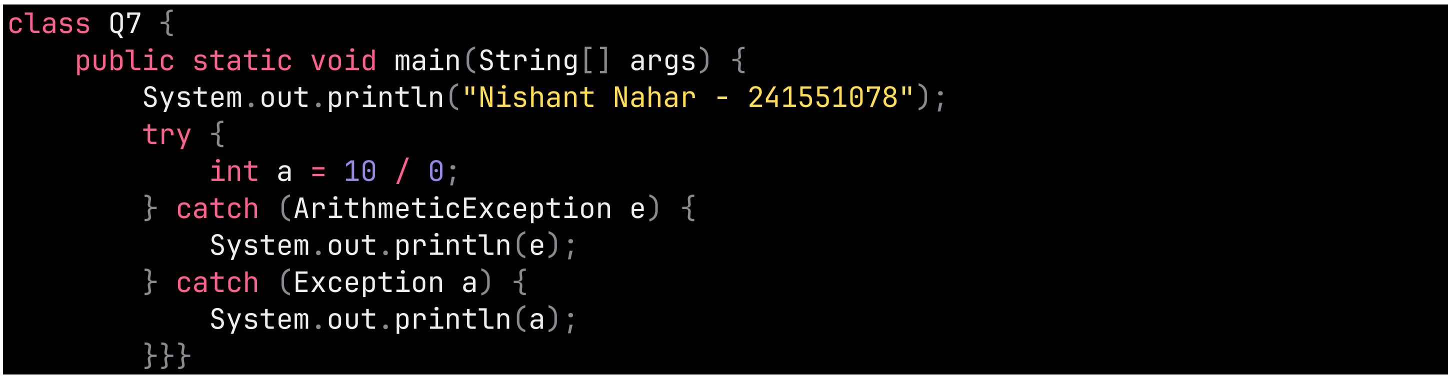
```
class Q6 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
            int[] a = new int[2];
            a[3] = 23 / 0;
        } catch (ArithmeticException e) {
            System.out.println(e);
        } catch (ArrayIndexOutOfBoundsException a) {
            System.out.println(a);
        }}}
```

OUTPUT



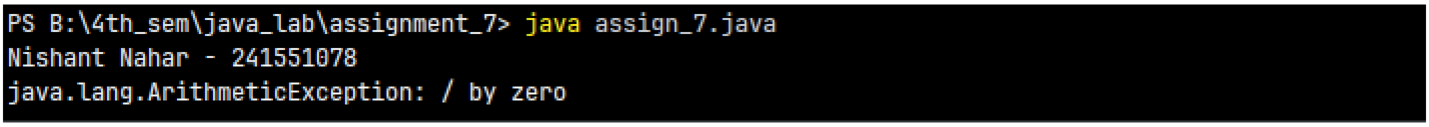
```
PS B:\4th_sem\java_lab\assignment_7> java assign_6.java
Nishant Nahar - 241551078
java.lang.ArithmeticException: / by zero
```

7. Write a program to demonstrate the importance of catch block order in exception handling.



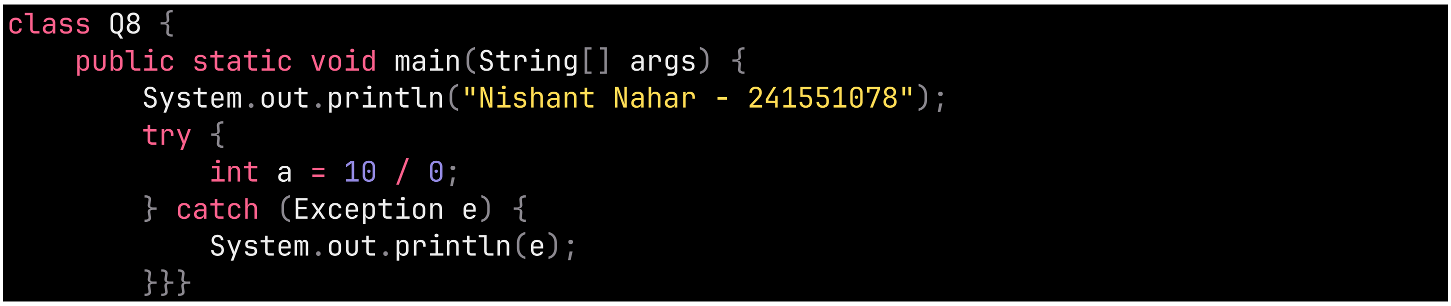
```
class Q7 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
            int a = 10 / 0;
        } catch (ArithmeticException e) {
            System.out.println(e);
        } catch (Exception a) {
            System.out.println(a);
        }}}
```

OUTPUT



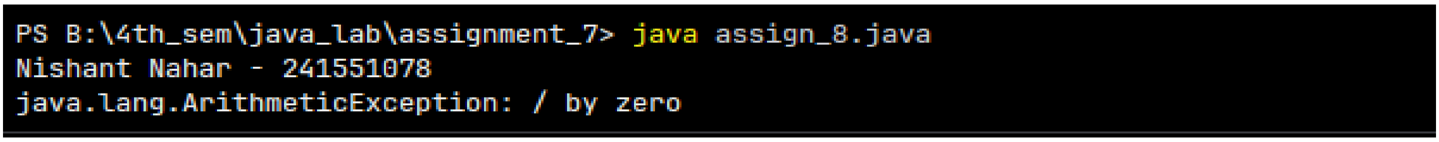
```
PS B:\4th_sem\java_lab\assignment_7> java assign_7.java
Nishant Nahar - 241551078
java.lang.ArithmeticException: / by zero
```

8. Write a program that catches a parent exception class instead of child exceptions and analyze the effect.



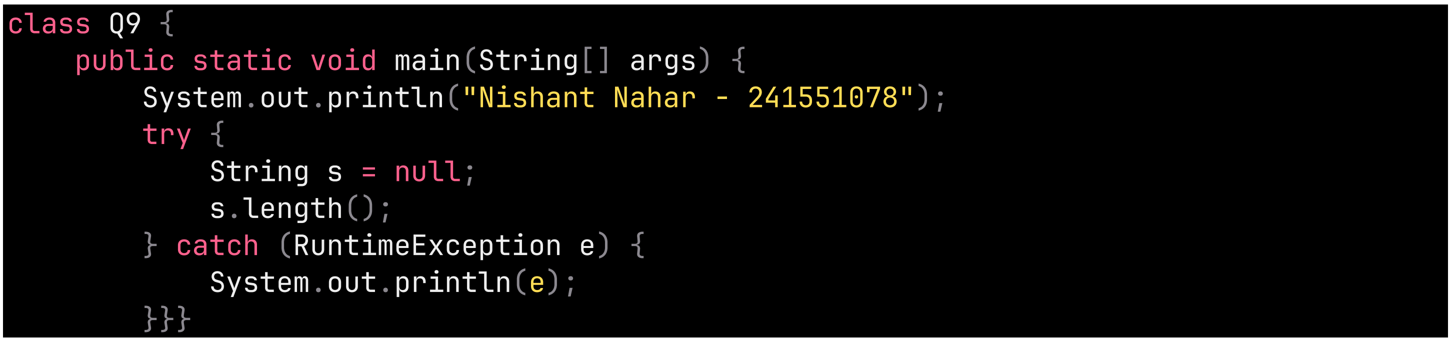
```
class Q8 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
            int a = 10 / 0;
        } catch (Exception e) {
            System.out.println(e);
        }}}
```

OUTPUT



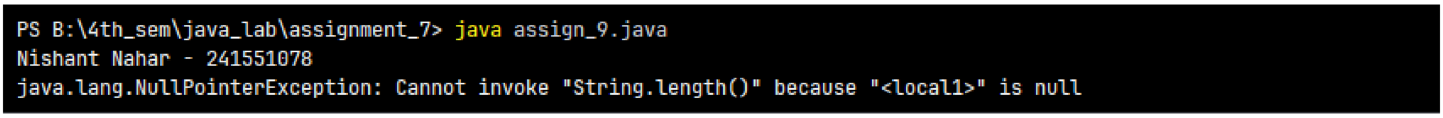
```
PS B:\4th_sem\java_lab\assignment_7> java assign_8.java
Nishant Nahar - 241551078
java.lang.ArithmeticException: / by zero
```

9. Write a program that shows how exception hierarchy works using built-in exceptions.



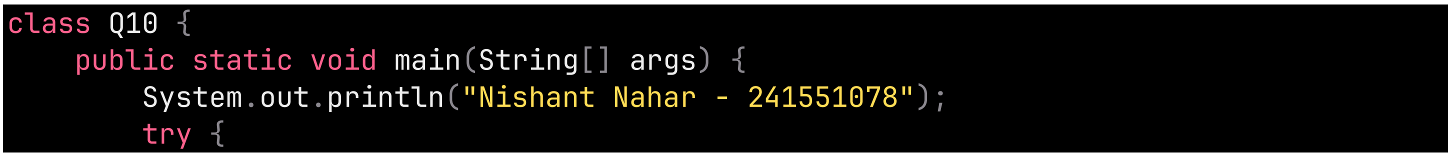
```
class Q9 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
            String s = null;
            s.length();
        } catch (RuntimeException e) {
            System.out.println(e);
        }}}
```

OUTPUT

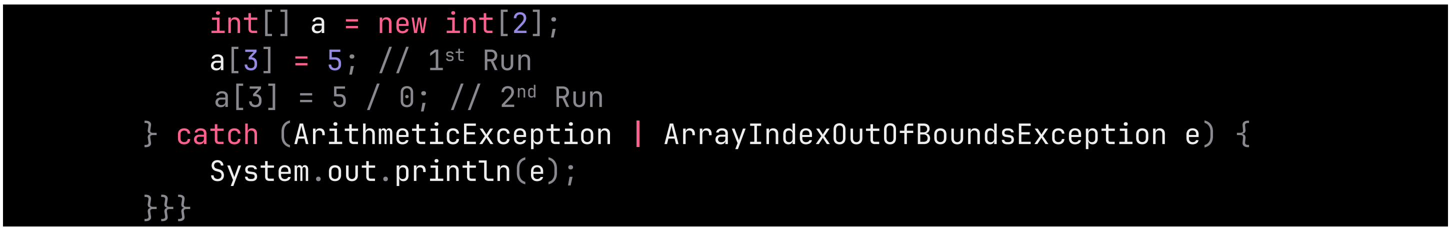


```
PS B:\4th_sem\java_lab\assignment_7> java assign_9.java
Nishant Nahar - 241551078
java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null
```

10. Write a program to demonstrate multi-catch (|) introduced in Java 7.

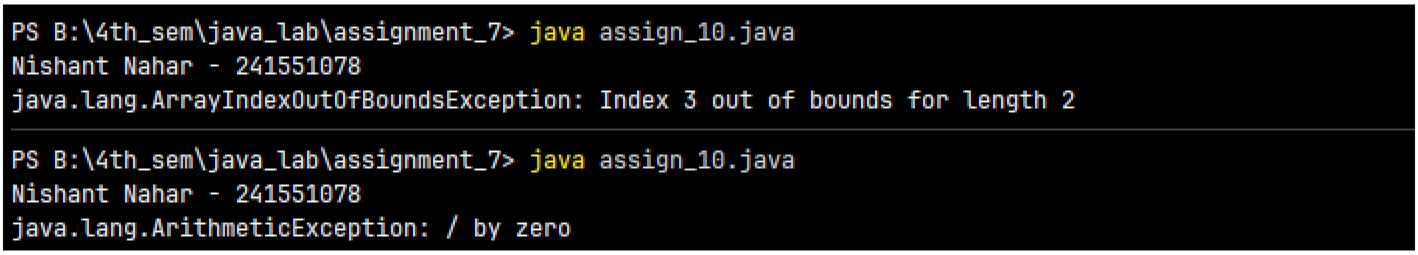


```
class Q10 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
```



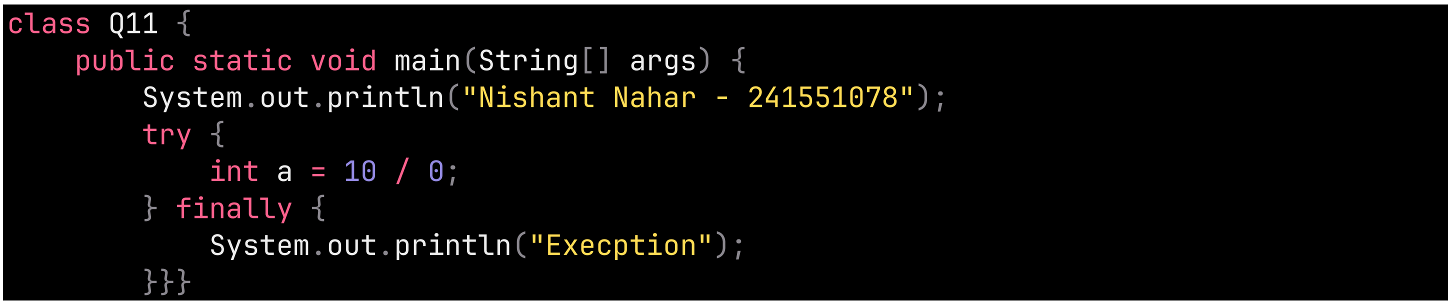
```
            int[] a = new int[2];
            a[3] = 5; // 1st Run
            a[3] = 5 / 0; // 2nd Run
        } catch (ArithmeticException | ArrayIndexOutOfBoundsException e) {
            System.out.println(e);
        }}}
```

OUTPUT



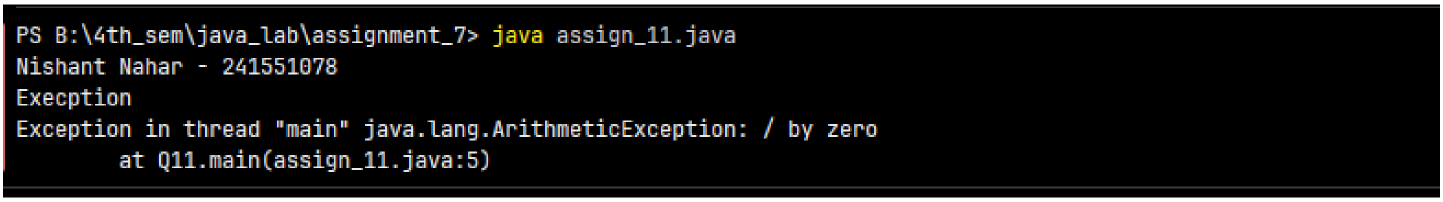
```
PS B:\4th_sem\java_lab\assignment_7> java assign_10.java
Nishant Nahar - 241551078
java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 2

PS B:\4th_sem\java_lab\assignment_7> java assign_10.java
Nishant Nahar - 241551078
java.lang.ArithmeticException: / by zero
```

11. Write a program to show that the finally block executes even when an exception occurs.



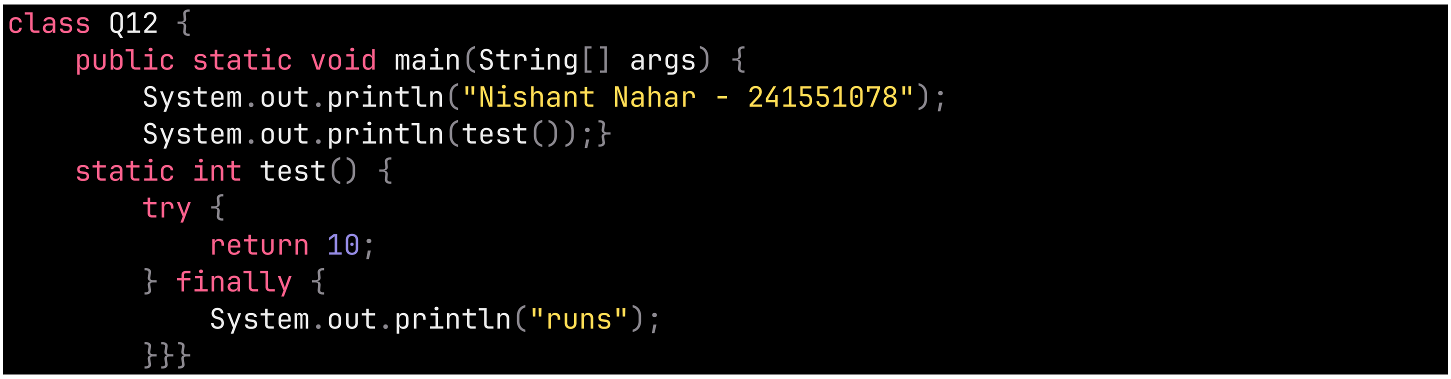
```
class Q11 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
            int a = 10 / 0;
        } finally {
            System.out.println("Execption");
        }}}
```

OUTPUT



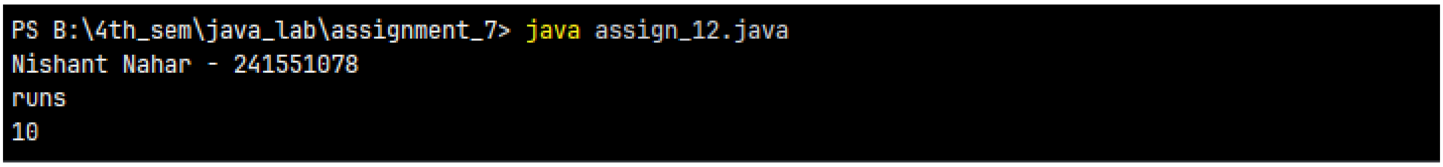
```
PS B:\4th_sem\java_lab\assignment_7> java assign_11.java
Nishant Nahar - 241551078
Execption
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Q11.main(assign_11.java:5)
```

12. Write a program to demonstrate that finally executes even if a return statement is present.



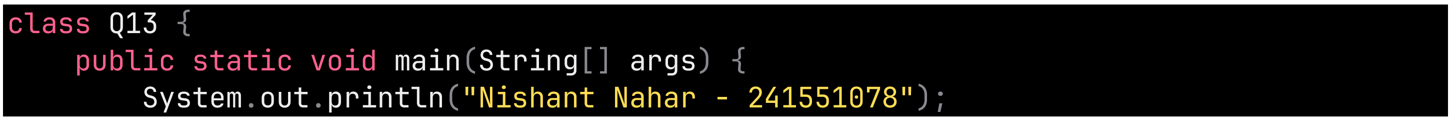
```
class Q12 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        System.out.println(test());}
    static int test() {
        try {
            return 10;
        } finally {
            System.out.println("runs");
        }}}
```

OUTPUT

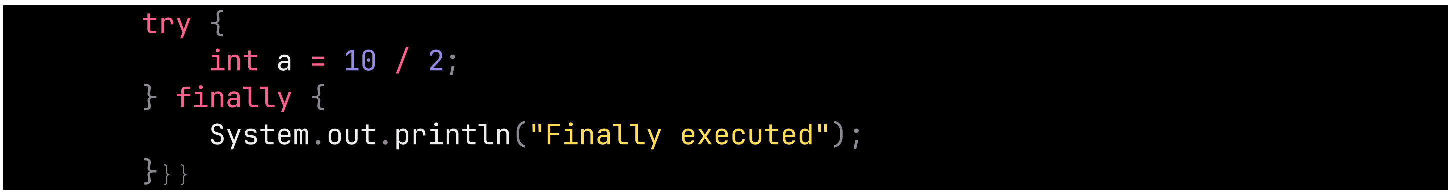


```
PS B:\4th_sem\java_lab\assignment_7> java assign_12.java
Nishant Nahar - 241551078
runs
10
```

13. Write a program to show that finally executes when no exception occurs.

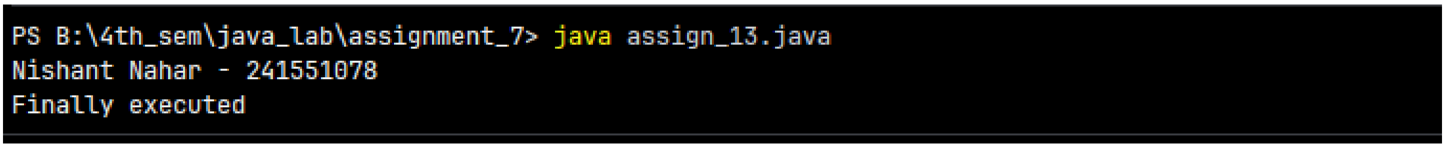


```
class Q13 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
```



```
        try {
            int a = 10 / 2;
        } finally {
            System.out.println("Finally executed");
        }}}
```

OUTPUT

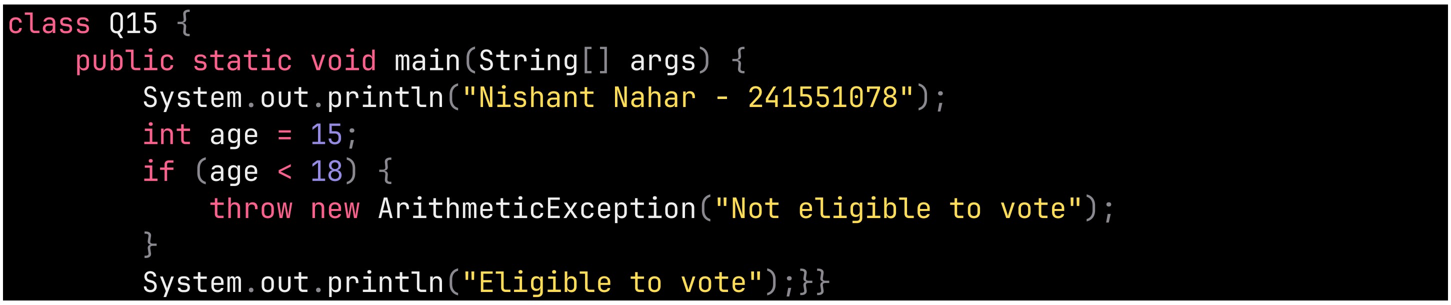


```
PS B:\4th_sem\java_lab\assignment_7> java assign_13.java
Nishant Nahar - 241551078
Finally executed
```

14. Write a program where the finally block does not execute and explain the reason.

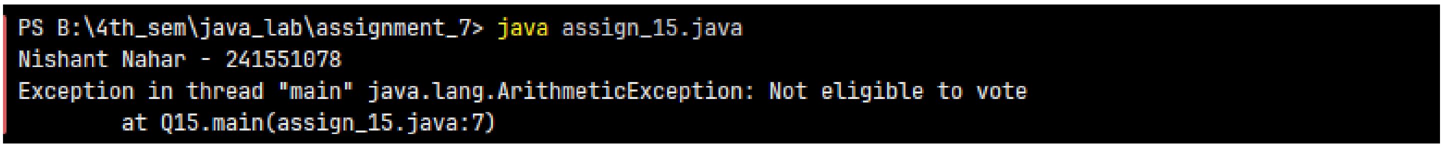
OUTPUT

15. Write a program that explicitly throws an exception using the throw keyword.



```
class Q15 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        int age = 15;
        if (age < 18) {
            throw new ArithmeticException("Not eligible to vote");
        }
        System.out.println("Eligible to vote");}}
```

OUTPUT

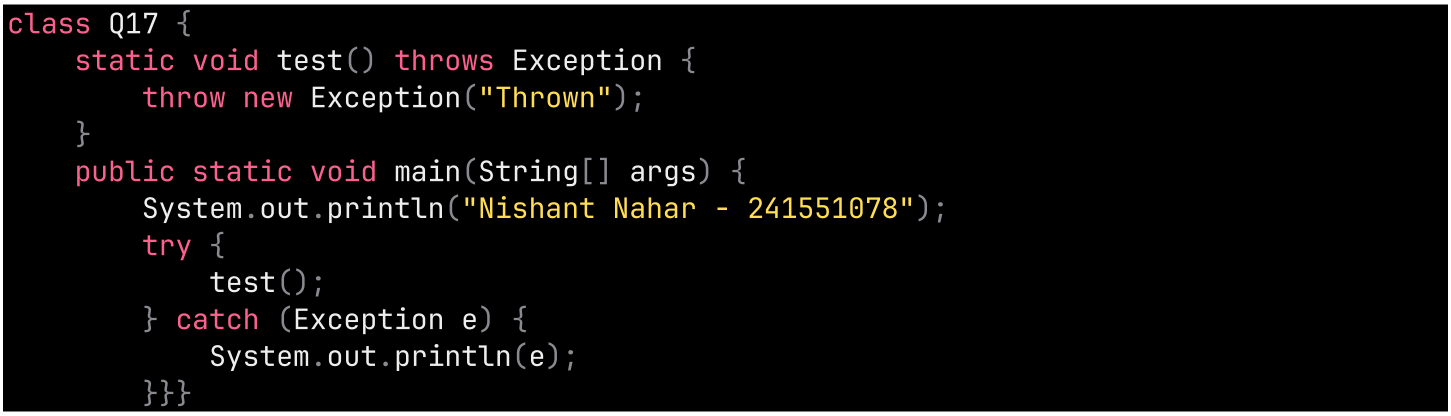


```
PS B:\4th_sem\java_lab\assignment_7> java assign_15.java
Nishant Nahar - 241551078
Exception in thread "main" java.lang.ArithmeticException: Not eligible to vote
        at Q15.main(assign_15.java:7)
```

16. Write a program that uses throws to declare a checked exception in a method.

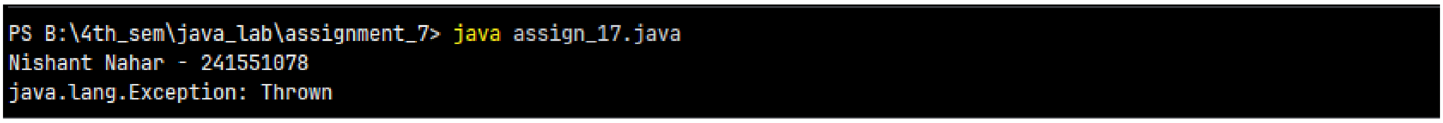
OUTPUT

17. Write a program demonstrating the difference between throw and throws.



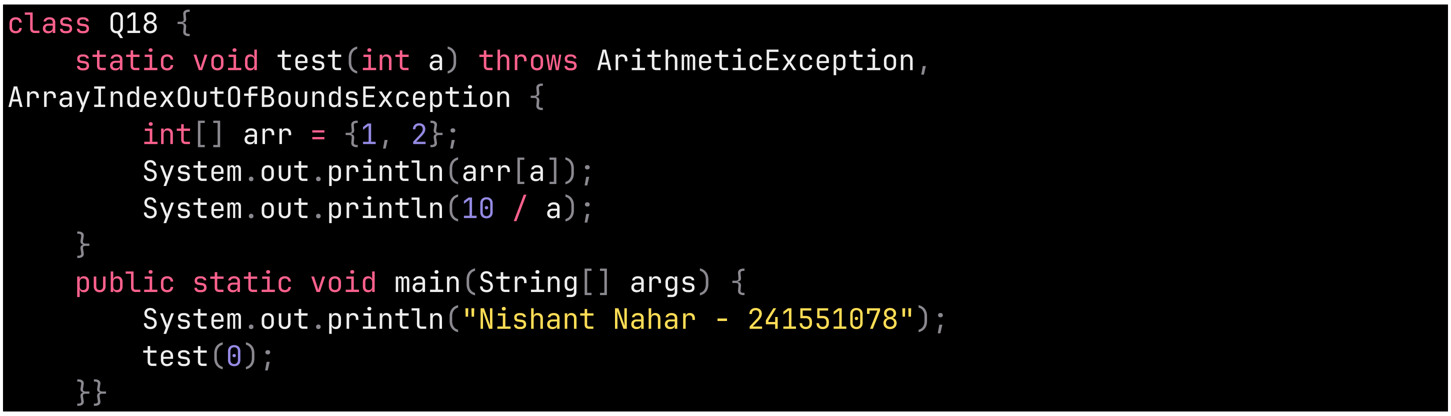
```
class Q17 {
    static void test() throws Exception {
        throw new Exception("Thrown");
    }
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        try {
            test();
        } catch (Exception e) {
            System.out.println(e);
        }}}
```

OUTPUT



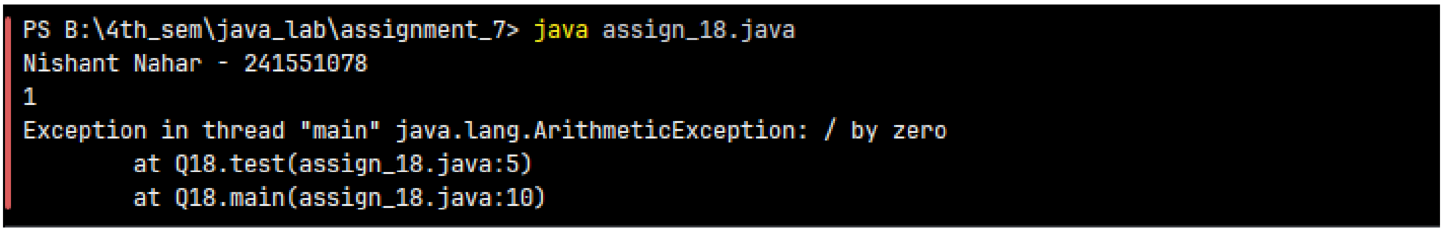
```
PS B:\4th_sem\java_lab\assignment_7> java assign_17.java
Nishant Nahar - 241551078
java.lang.Exception: Thrown
```

18. Write a program where a method throws multiple exceptions using throws.



```
class Q18 {
    static void test(int a) throws ArithmeticException,
ArrayIndexOutOfBoundsException {
        int[] arr = {1, 2};
        System.out.println(arr[a]);
        System.out.println(10 / a);
    }
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        test(0);
    }}
```

OUTPUT



```
PS B:\4th_sem\java_lab\assignment_7> java assign_18.java
Nishant Nahar - 241551078
1
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Q18.test(assign_18.java:5)
        at Q18.main(assign_18.java:10)
```

19. Create a custom checked exception and use it in a validation program.

OUTPUT

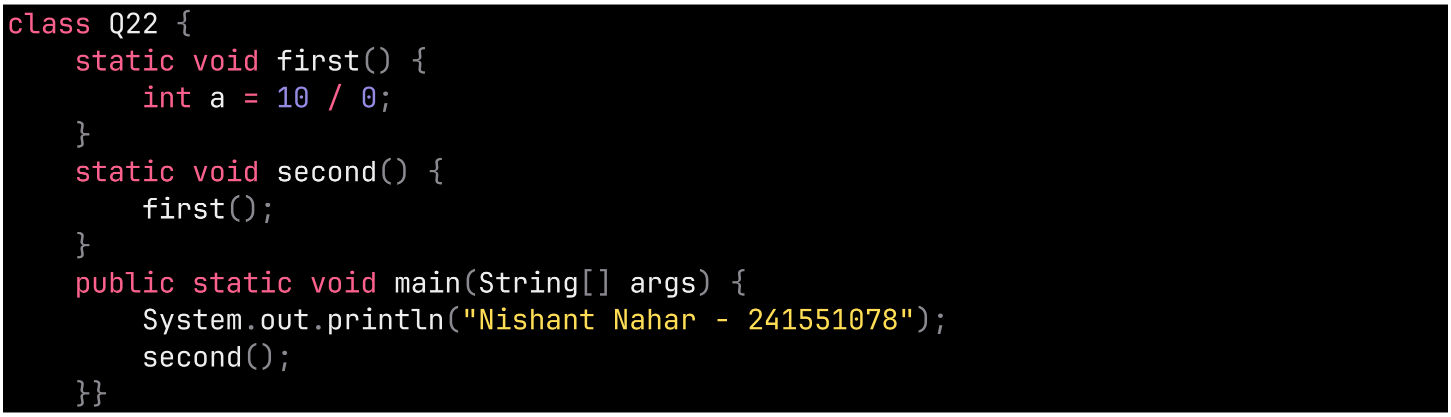
20. Create a custom unchecked exception for a real-world scenario (e.g., insufficient balance).

OUTPUT

21. Write a program that passes a message to a user-defined exception.

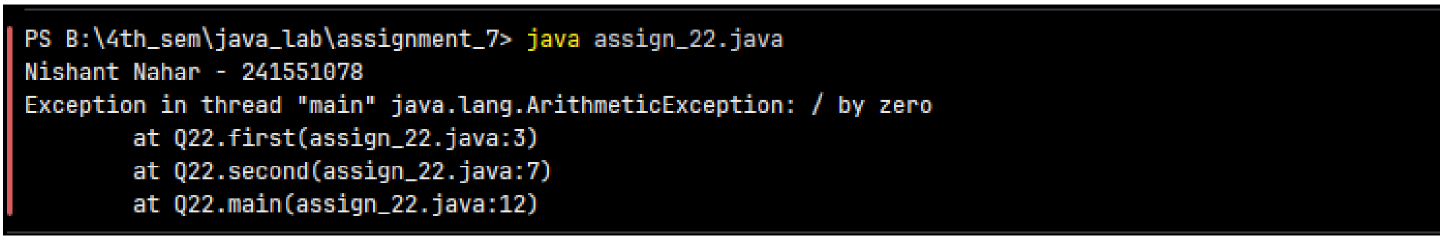
OUTPUT

22. Write a program to demonstrate exception propagation across multiple method calls.



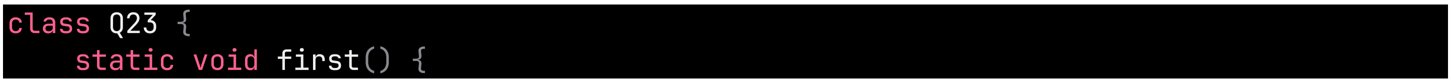
```
class Q22 {
    static void first() {
        int a = 10 / 0;
    }
    static void second() {
        first();
    }
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        second();
    }}
```

OUTPUT

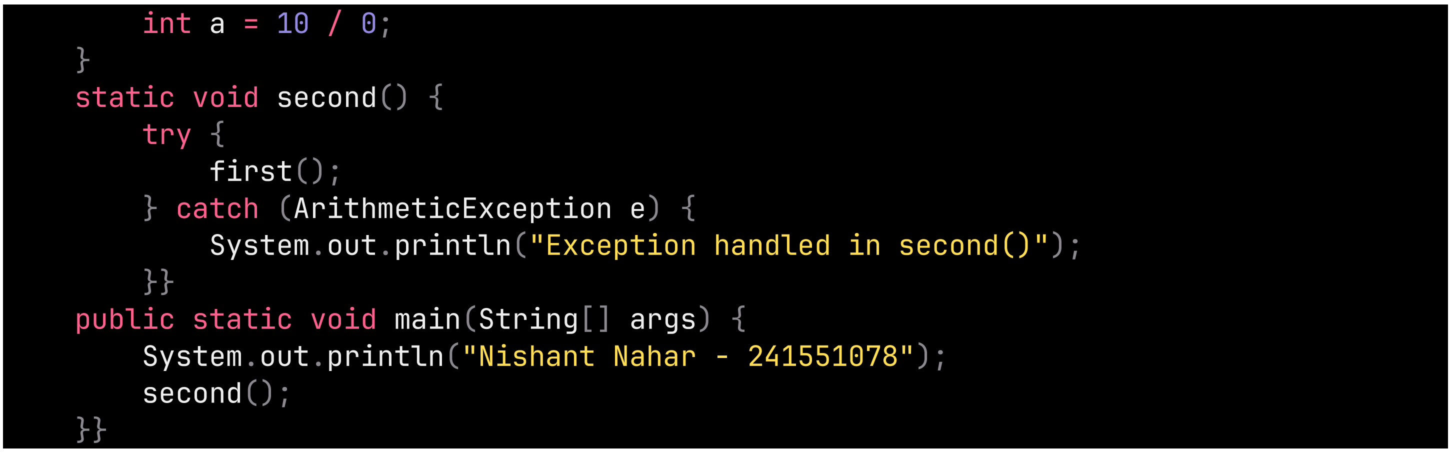


```
PS B:\4th_sem\java_lab\assignment_7> java assign_22.java
Nishant Nahar - 241551078
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Q22.first(assign_22.java:3)
        at Q22.second(assign_22.java:7)
        at Q22.main(assign_22.java:12)
```

23. Modify the above program to handle the exception at an intermediate method.

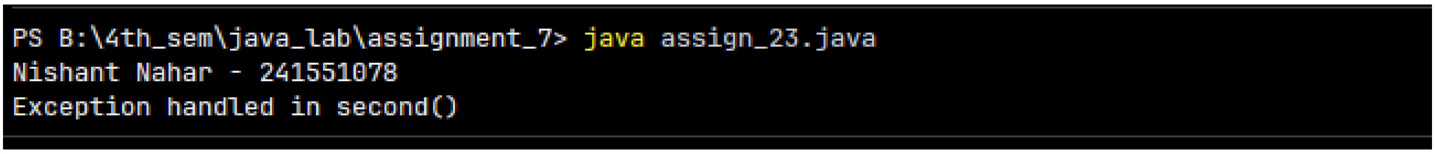


```
class Q23 {
    static void first() {
```



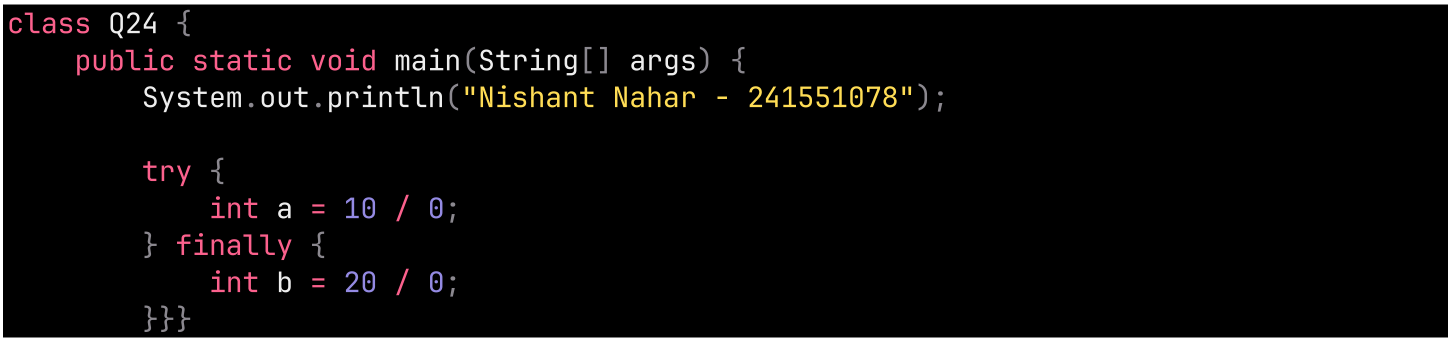
```
        int a = 10 / 0;
    }
    static void second() {
        try {
            first();
        } catch (ArithmeticException e) {
            System.out.println("Exception handled in second()");
        }}
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");
        second();
    }}
```

OUTPUT



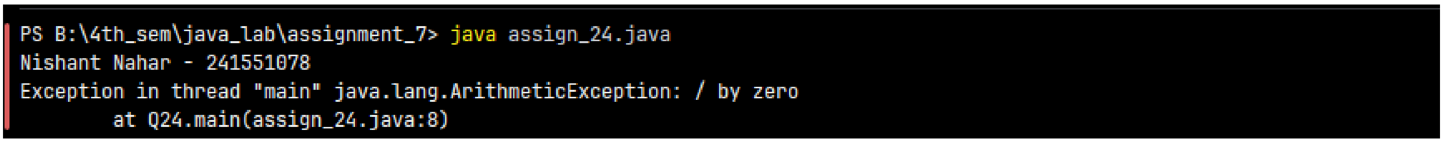
```
PS B:\4th_sem\java_lab\assignment_7> java assign_23.java
Nishant Nahar - 241551078
Exception handled in second()
```

24. Write a program where an exception is thrown in both try and finally blocks. Observe the output.



```
class Q24 {
    public static void main(String[] args) {
        System.out.println("Nishant Nahar - 241551078");

        try {
            int a = 10 / 0;
        } finally {
            int b = 20 / 0;
        }}}
```

OUTPUT



```
PS B:\4th_sem\java_lab\assignment_7> java assign_24.java
Nishant Nahar - 241551078
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Q24.main(assign_24.java:8)
```

25. Write a program to demonstrate rethrowing an exception after partial handling.
26. Write a program to show how try-with-resources works with file handling.
27. Write a program to demonstrate suppressed exceptions in try-with-resources.
28. Write a program to demonstrate custom exception chaining using initCause().
29. Write a program to show the difference between checked and unchecked exceptions at compile time.
30. Write a program that demonstrates why catching Throwable is discouraged.

P.S – The remaining Question were not done till now