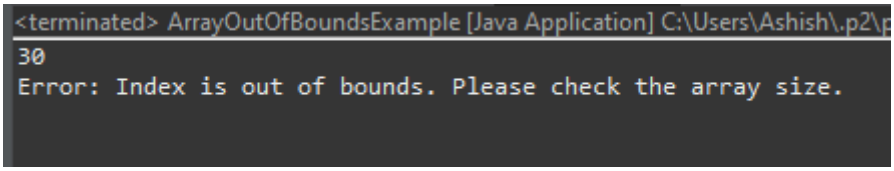
Lab 7

1. Write a program that tries to access an element outside the bounds of an array and handles the ArrayIndexOutOfBoundsException by printing a user-friendly message.

Code:

```
package hellow;

public class ArrayOutOfBoundsExample {
  public static void main(String[] args) {
    int[] numbers = {10, 20, 30, 40};

    try {
      // Accessing an element within the bounds
      System.out.println(numbers[2]); // This will print 30

      // Accessing an element outside the bounds
      System.out.println(numbers[5]); // This will throw an exception
    } catch (ArrayIndexOutOfBoundsException e) {
      System.out.println("Error: Index is out of bounds. Please check the array
size.");
    }
  }
}
```

Output :

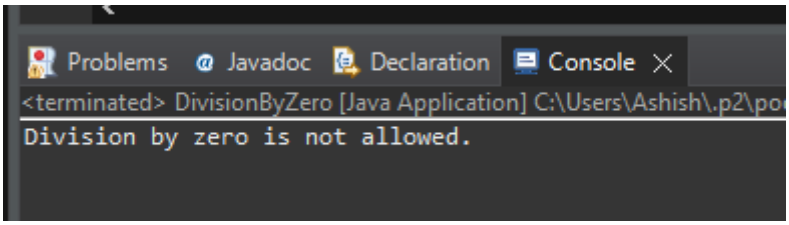


2. Write a program that attempts to divide a number by zero and handles the ArithmeticException by printing a message that division by zero is not allowed.

Code:

```
package hellow;

public class DivisionByZero {
  public static void main(String[] args) {
    try {
      // Attempting to divide by zero
      int result = 10 / 0;
    } catch (ArithmeticException e) {
      // Handling the ArithmeticException
      System.out.println("Division by zero is not allowed.");
    }
  }
}
```

Output :

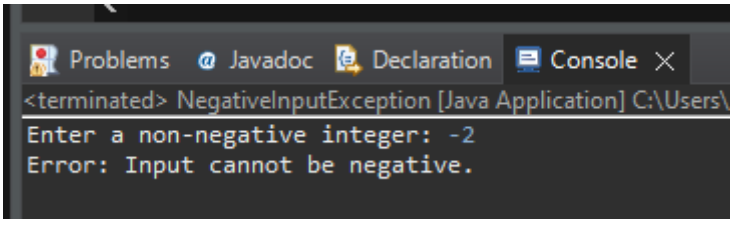


3. Write a Java program that reads an integer input from the user and throws an IllegalArgumentException if the input is negative.
Display an appropriate message when the exception is caught.

Code:

```
package hellow;

import java.util.Scanner;

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

    try {
      System.out.print("Enter a non-negative integer: ");
      int number = scanner.nextInt();

      if (number < 0) {
        throw new IllegalArgumentException("Input cannot be negative.");
      }

      // Rest of your code to process the positive number
      System.out.println("You entered: " + number);

    } catch (IllegalArgumentException e) {
      System.out.println("Error: " + e.getMessage());
    }
  }
}
```

Output :

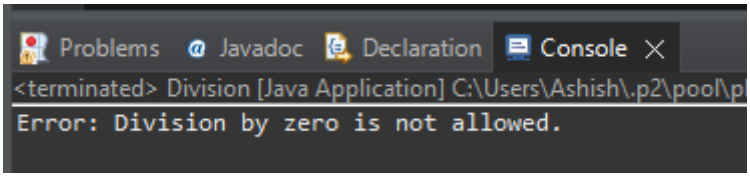


4. Create a Java method that divides two numbers and declares that it throws an ArithmeticException. Handle the exception in the main method.

Code:

```
package hellow;
public class Division {

  public static int divide(int dividend, int divisor) throws ArithmeticException {
    return dividend / divisor;
  }

  public static void main(String[] args) {
    int numerator = 10;
    int denominator = 0;

    try {
      int result = divide(numerator, denominator);
      System.out.println("Result: " + result);
    } catch (ArithmeticException e) {
      System.out.println("Error: Division by zero is not allowed.");
    }
  }
}
```

Output :



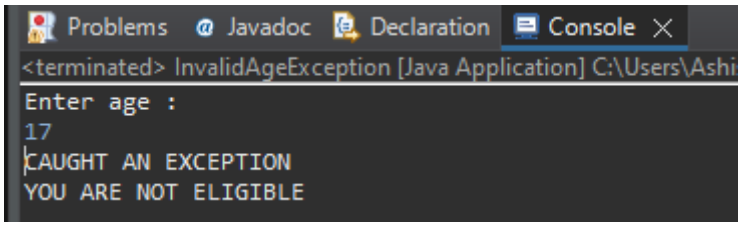
5. Define a custom exception called InvalidAgeException. Write a Java program that throws this exception if the age provided is less than 18. Handle the exception and display an appropriate message

Code:

```
package hellow;
import java.util.*;
class AgeException extends Exception {
  AgeException(String message) {
    super(message);
  }
}

public class InvalidAgeException {
  public static void main(String[] args) {
    int age;
    Scanner sc = new Scanner(System.in);

    try {
```

```
      System.out.println("Enter age : ");
      age = sc.nextInt();

      if (age < 18) {
        throw new AgeException("YOU ARE NOT ELIGIBLE");
      } else {
        System.out.println("You are eligible for vote");
      }

    } catch (AgeException e) {
      System.out.println("CAUGHT AN EXCEPTION");
      System.out.println(e.getMessage());
    }
  }
}
```

Output :



6. Write a Java program that has a method to validate a user's email address. The method should throw a custom exception Invalid Email Exception if the email does not contain @ and.. Handle the exception in the main method.

Code:

```
package hellow;

import java.util.Scanner;

class InvalidEmailException extends Exception {
  public InvalidEmailException(String message) {
    super(message);
  }
}

public class EmailValidatorr {
  public static void validateEmail(String email) throws InvalidEmailException {
    if (!email.contains("@") || !email.contains(".")) {
      throw new InvalidEmailException("Invalid email format");
    }
  }

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter your email: ");
    String email = scanner.nextLine();
```

```
    try {
      validateEmail(email);
      System.out.println("Valid email address");
    } catch (InvalidEmailException e) {
      System.out.println("Error: " + e.getMessage());
    }
  }
}
```

Output :