Exception handing

Question 01

Write a Java program that takes two integers as input and performs division on them. Implement exception handling to catch and handle the ArithmeticException that occurs when dividing by zero.

Print an appropriate error message if the denominator is zero.

Answers

```
public class DivideTwoIntegers {

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

        System.out.print("Enter the first integer: ");
        int numerator = scanner.nextInt();

        System.out.print("Enter the second integer: ");
        int denominator = scanner.nextInt();

        try {
            int result = numerator / denominator;
            System.out.println("The result is: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Division by zero error.");
        }
    }
}
```

Question 02

Write a Java program that creates an array of integers and attempts to access an index that is out of bounds. Implement exception handling to catch and handle the ArrayIndexOutOfBoundsException.

Print an appropriate error message if an invalid index is accessed.

Answers

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

        try {
            // Create an array of integers
            int[] numbers = { 10, 20, 30, 40, 50 };

            System.out.print("Enter an index to access: ");
            int index = scanner.nextInt();
            int element = numbers[index];

            System.out.println("Element at index " + index + ": " + element);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Invalid index. The index is out of bounds.");
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter a valid index.");
        } finally {
            scanner.close();
        }
    }
}
```

Question 03

Write a Java program that attempts to read a file that does not exist. Implement exception handling to catch and handle the FileNotFoundException.

Print an appropriate error message if the file is not found.

Answers

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class FileReadProgram {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
            System.out.print("Enter the file name: ");
            String fileName = scanner.nextLine();

            File file = new File(fileName);

            if (!file.exists()) {
                throw new FileNotFoundException("Error: The file '" + fileName + "' does not exist.");
            }

            Scanner fileScanner = new Scanner(file);
            while (fileScanner.hasNextLine()) {
                String line = fileScanner.nextLine();
                System.out.println(line);
            }
            fileScanner.close();
        } catch (FileNotFoundException e) {
            System.out.println(e.getMessage());
        } catch (Exception e) {
            System.out.println("Error: An unexpected error occurred while reading the file.");
        } finally {
            scanner.close();
        }
```

```
    }
}
```