**1)** import java.util.Scanner;

public class DivisionExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Read the first integer

System.out.print("Enter the first number: ");

int num1 = scanner.nextInt();

// Read the second integer

System.out.print("Enter the second number: ");

int num2 = scanner.nextInt();

try {

// Perform the division

int result = num1 / num2;

System.out.println("The result of division is: " + result);

} catch (ArithmeticException e) {

// Handle the case where division by zero occurs

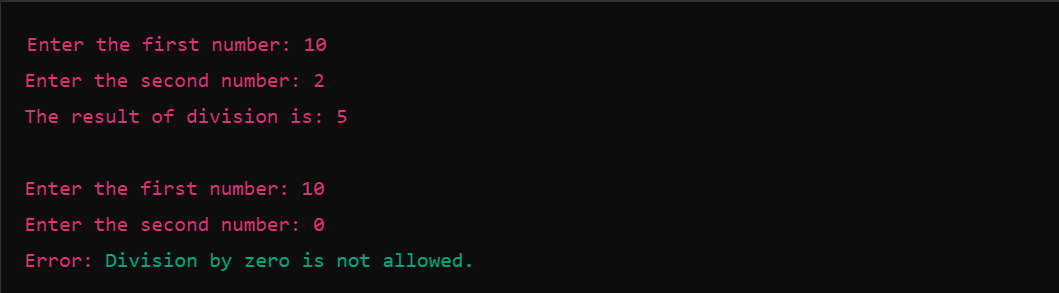
System.out.println("Error: Division by zero is not allowed.");

} finally {

scanner.close(); // Close the scanner resource

}

}

} 

**2) ArrayIndexOutOfBoundsException**:

public class ArrayIndexOutOfBoundsExample {

public static void main(String[] args) {

int[] numbers = {1, 2, 3}; // Array with 3 elements

try {

// Trying to access an invalid index (index 5 is out of bounds for this array)

System.out.println(numbers[5]);

} catch (ArrayIndexOutOfBoundsException e) {

// Handling the exception

System.out.println("Error: Array index is out of bounds!");

}

}

}



**StringIndexOutOfBoundsException**:

public class StringIndexOutOfBoundsExample {

public static void main(String[] args) {

String text = "Hello"; // String with 5 characters

try {

// Trying to access an invalid index (index 10 is out of bounds for this string)

System.out.println(text.charAt(10));

} catch (StringIndexOutOfBoundsException e) {

// Handling the exception

System.out.println("Error: String index is out of bounds!");

}

}

}



**3) Step 1: Create the Custom Exception - InvalidAgeException**

// Custom exception class extending Exception

class InvalidAgeException extends Exception {

public InvalidAgeException(String message) {

super(message); // Call the superclass (Exception) constructor with the message

}

}

**Step 2: Implementing the Program with Exception Handling**

import java.util.Scanner;

public class InvalidAgeExceptionExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user for age

System.out.print("Enter your age: ");

int age = scanner.nextInt();

try {

// Check if the age is less than 18

if (age < 18) {

// Throw the custom exception

throw new InvalidAgeException("Error: Age must be 18 or older.");

} else {

System.out.println("You are eligible.");

}

} catch (InvalidAgeException e) {

// Catch and handle the custom exception

System.out.println(e.getMessage()); // Display the exception message

} finally {

scanner.close(); // Close the scanner resource

}

}

}

**Case 1: When age is valid (>= 18):**

****

**Case 2: When age is invalid (< 18):**

****

**4**) import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class FileReadingExample {

public static void main(String[] args) {

// Specify the file path (use a non-existent file for testing)

String fileName = "example.txt"; // Change to a valid file name for testing

// Try to read the file

try {

// Create a File object for the specified file

File file = new File(fileName);

// Create a Scanner object to read the file

Scanner fileScanner = new Scanner(file);

// Read and print each line of the file

while (fileScanner.hasNextLine()) {

System.out.println(fileScanner.nextLine());

}

// Close the file scanner

fileScanner.close();

} catch (FileNotFoundException e) {

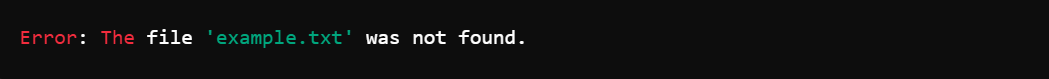
// Handle the case where the file is not found

System.out.println("Error: The file '" + fileName + "' was not found.");

}

}

}



**5)** import java.util.ArrayList;

public class ArrayListExample {

public static void main(String[] args) {

// Create an ArrayList of Strings

ArrayList<String> fruits = new ArrayList<>();

// Add some elements to the ArrayList

fruits.add("Apple");

fruits.add("Banana");

fruits.add("Orange");

fruits.add("Mango");

// Print the ArrayList before removing elements

System.out.println("ArrayList before removing elements: " + fruits);

// Remove all elements from the ArrayList

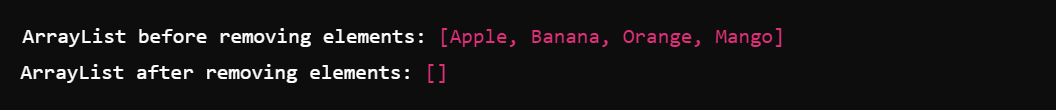
fruits.clear();

// Print the ArrayList after removing elements

System.out.println("ArrayList after removing elements: " + fruits);

}

}



**6**) import java.util.Map;

import java.util.TreeMap;

public class EmployeeTreeMap {

public static void main(String[] args) {

// Create a TreeMap to store employee IDs and names

TreeMap<Integer, String> employeeMap = new TreeMap<>();

// Add some employee IDs and names to the TreeMap

employeeMap.put(101, "John");

employeeMap.put(102, "Alice");

employeeMap.put(103, "Bob");

employeeMap.put(104, "David");

employeeMap.put(105, "Eve");

// Print employee names in alphabetical order

System.out.println("Employee names in alphabetical order:");

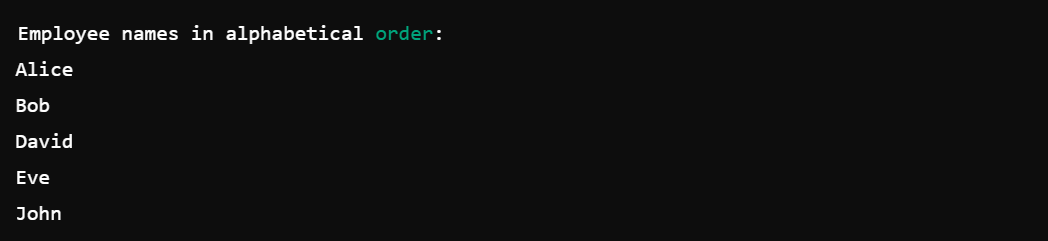
for (Map.Entry<Integer, String> entry : employeeMap.entrySet()) {

System.out.println(entry.getValue());

}

}

}



**7**) import java.util.ArrayList;

import java.util.List;

public class ListToArrayExample {

public static void main(String[] args) {

// Create a List of Strings

List<String> fruitList = new ArrayList<>();

fruitList.add("Apple");

fruitList.add("Banana");

fruitList.add("Cherry");

fruitList.add("Date");

// Convert List to Array using toArray() method

String[] fruitArray = fruitList.toArray(new String[0]);

// Print the Array

System.out.println("Array elements:");

for (String fruit : fruitArray) {

System.out.println(fruit);

}

}

}

