Q.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:

public class ArrayOutOfBoundsDemo {

public static void main(String[] args) {

// Define an array with 5 elements

int[] array = { 1, 2, 3, 4, 5 };

try {

// Attempt to access an element outside the bounds of the array

int element = array[10];

System.out.println("Element at index 10: " + element);

} catch (ArrayIndexOutOfBoundsException e) {

// Handle the exception by printing a user-friendly message

System.out.println("Error: Attempted to access an index that is out of bounds.");

}

// Continue program execution after handling the exception

System.out.println("Program continues after handling the exception.");

}

}

OUTPUT:



Q.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:

public class DivisionByZeroDemo {

public static void main(String[] args) {

// Define numerator and denominator

int numerator = 10;

int denominator = 0;

try {

// Attempt to divide by zero

int result = numerator / denominator;

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

} catch (ArithmeticException e) {

// Handle the exception by printing a user-friendly message

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

}

// Continue program execution after handling the exception

System.out.println("Program continues after handling the exception.");

}

}

OUTPUT:



Q.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:

import java.util.Scanner;

public class NegativeInputChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter an integer: ");

try {

int input = scanner.nextInt();

checkForNegative(input);

System.out.println("You entered: " + input);

} catch (IllegalArgumentException e) {

System.out.println(e.getMessage());

} finally {

scanner.close();

}

}

public static void checkForNegative(int number) {

if (number < 0) {

throw new IllegalArgumentException("Negative numbers are not allowed.");

}

}

}

OUTPUT:





Q.4 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:

import java.security.InvalidAlgorithmParameterException;

import java.util.Scanner;

public class AgeVerification {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in); // Create a Scanner object to read input

System.out.print("Please enter your age: "); // Prompt user to enter their age

int age = scanner.nextInt(); // Read the user's input

try {

checkAge(age); // Check if the age is valid

System.out.println("Your age is valid."); // Print message if age is valid

} catch (InvalidAlgorithmParameterException e) {

// Handle the custom exception by printing a user-friendly message

System.out.println("Error: " + e.getMessage());

}

// Continue with the rest of the program

System.out.println("Program continues after handling the exception.");

scanner.close(); // Close the scanner to free resources

}

// Method to check the age

public static void checkAge(int age) throws InvalidAlgorithmParameterException {

if (age < 18) {

// Throw custom exception if age is less than 18

throw new InvalidAlgorithmParameterException("Age must be 18 or older.");

}

}

}

OUTPUT:

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Q.5 Write a Java program that has a method to validate a user's email address. The method should throw a custom exception InvalidEmailException if the email does not contain @ and .. Handle the exception in the main method.

CODE:

import java.util.Scanner;

public class EmailValidation {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in); // Create a Scanner object to read input

System.out.print("Please enter your email address: "); // Prompt user to enter their email address

String email = scanner.nextLine(); // Read the user's input

try {

validateEmail(email); // Validate the entered email address

System.out.println("Your email address is valid."); // Print message if email is valid

} catch (Exception e) {

System.out.println("Error: " + e.getMessage()); // Catch and handle any exception with a user-friendly message

}

System.out.println("Program continues after handling the exception."); // Continue with program execution

scanner.close(); // Close the scanner to free resources

}

// Method to validate the email address

public static void validateEmail(String email) throws Exception {

if (!email.contains("@") || !email.contains(".")) {

throw new Exception("Email address must contain '@' and '.'"); // Throw exception if email format is invalid

}

}

}

OUTPUT:

A close-up of a white background

Description automatically generated

A close-up of a white background

Description automatically generated