**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.**

**PROGRAM:**

**package** Lab7;

**public** **class** ArrayOutOfBoundDemo {

**public** **static** **void** main(String[] args) {

// Creating an array of size 5

**int** numArr[] = **new** **int**[5];

// Assigning values to array indexes

numArr[0] = 1;

numArr[1] = 2;

numArr[2] = 3;

numArr[3] = 4;

numArr[4] = 5;

**try** {

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

System.***out***.println(numArr[6]);

}

**catch** (ArrayIndexOutOfBoundsException e) {

// Printing a user-friendly message with valid index information

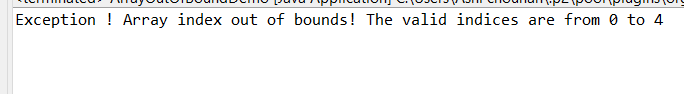
System.***out***.println("Exception ! Array index out of bounds! The valid indices are from 0 to " + (numArr.length - 1) );

}

}

}

**OUTPUT:**

****

1. **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.**

**PROGRAM:**

**package** Lab7;

**public** **class** ZeroDivisionExceptionDemo {

**public** **static** **void** main(String[] args) {

**int** num1 = 10;

**int** num2 = 0;

**try** {

// Attempting to divide by zero

System.***out***.println(num1 / num2);

}

**catch** (ArithmeticException e) {

// Printing a user-friendly message

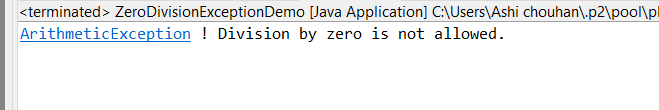
System.***out***.println("ArithmeticException ! Division by zero is not allowed.");

}

}

}

**OUTPUT:**

****

1. **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.**

**PROGRAM:**

**package** Lab7;

**import** java.util.Scanner;

**public** **class** IllegalArgumentDemo {

**public** **static** **void** main(String[] args) {

// Scanner class object is created

Scanner scanner = **new** Scanner(System.***in***);

**try** {

// Taking integer input from user

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

**int** input = scanner.nextInt();

// Checking if input is negative

**if** (input < 0) {

**throw** **new** IllegalArgumentException("Input cannot be negative.");

}

**else** {

// Display the valid input

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

}

} **catch** (IllegalArgumentException e) {

// Printing message when exception is caught

System.***out***.println("Exception caught ! " + e.getMessage());

} **finally** {

// Close the scanner

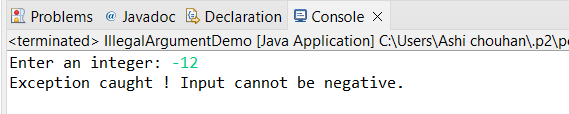
scanner.close();

}

}

}

**OUTPUT:**

****

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

**PROGRAM:**

**package** Lab7;

**public** **class** ArithmeticExceptionDemo {

// Method to divide two numbers

**public** **static** **double** divide(**double** num1, **double** num2) **throws** ArithmeticException {

**if** (num2 == 0) {

// Error if num2 is 0

**throw** **new** ArithmeticException("Cannot divide by zero.");

}

**return** num1 / num2; // Return result

}

**public** **static** **void** main(String[] args) {

**try** {

**double** num1 = 10; // Top number

**double** num2 = 0; // Bottom number

**double** result = *divide*(num1, num2); // Do division

System.***out***.println("Result: " + result); // Show result

} **catch** (ArithmeticException e) {

// Show error message

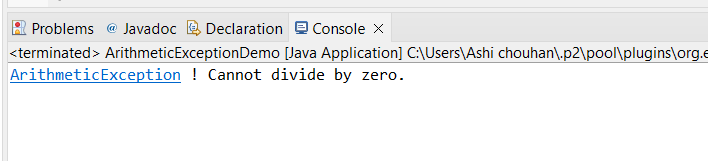
System.***out***.println("ArithmeticException ! " + e.getMessage());

}

}

}

**OUTPUT:**

****

1. **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**

**PROGRAM:**

**package** Lab7;

**class** InvalidAgeException **extends** Exception {

**public** InvalidAgeException(String message) {

**super**(message);

}

}

**public** **class** InvalidAgeExceptionDemo {

// Method to check age and throw InvalidAgeException if age is less than 18

**public** **static** **void** checkAge(**int** age) **throws** InvalidAgeException {

**if** (age < 18) {

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

}

**else** {

System.***out***.println("Age is valid.");

}

}

**public** **static** **void** main(String[] args) {

**try** {

**int** age = 16;

System.***out***.println("Age : " + age);

*checkAge*(age);

} **catch** (InvalidAgeException e) {

// Handle the custom exception

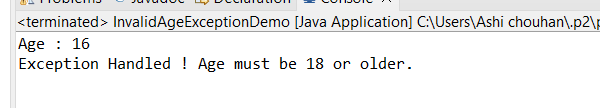
System.***out***.println("Exception Handled ! " + e.getMessage());

}

}

}

**OUTPUT:**

****

1. **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.**

**PROGRAM:**

**package** Lab7;

//Custom exception for invalid email

**class** InvalidEmailException **extends** Exception {

**public** InvalidEmailException(String msg) {

**super**(msg); // Set message for the exception

}

}

**public** **class** EmailValidatorDemo {

// Method to check if email is valid

**public** **static** **void** validateEmail(String email) **throws** InvalidEmailException {

**if** (email == **null** || !email.contains("@") || !email.contains(".")) {

// If email is missing @ or ., throw an exception

**throw** **new** InvalidEmailException("Email must include '@' and '.'.");

}

System.***out***.println("Email looks good."); // Email is valid

}

**public** **static** **void** main(String[] args) {

**try** {

String email = "userexample.com";

System.***out***.println("Email : " + email);

*validateEmail*(email); // Check if email is valid

} **catch** (InvalidEmailException e) {

// Print the error message

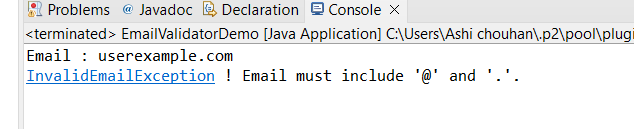
System.***out***.println("InvalidEmailException ! " + e.getMessage());

}

}

}

**OUTPUT:**

****