**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**.

**Program:**

**package** myPackage;

**public** **class** Array\_Ex {

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

// **TODO** Auto-generated method stub

// Initialize an array with some elements

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

**try** {

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

**int** index = 11; // Index that is out of bounds

System.***out***.println("Accessing element at index " + index + ": " + numbers[index]);

} **catch** (ArrayIndexOutOfBoundsException e) {

// Handle the exception and print a user-friendly message

System.***out***.println("Error: " + e.getMessage() + " is out of bounds for the array.");

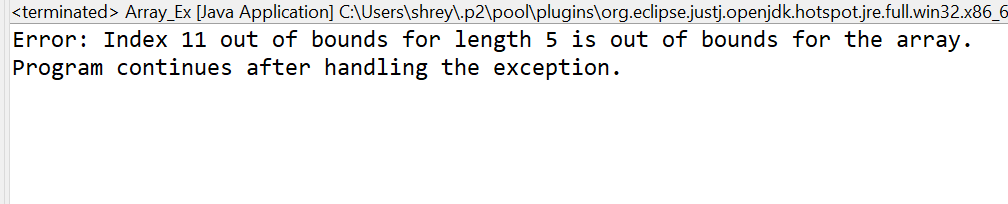
}

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**.

**Program:**

**package** myPackage;

**public** **class** Divide\_Demo {

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

// **TODO** Auto-generated method stub

**int** numerator = 10;

**int** denominator = 0;

**try** {

// Attempt to divide by zero

**int** result = numerator / denominator;

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

} **catch** (ArithmeticException e) {

// Handle the ArithmeticException

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

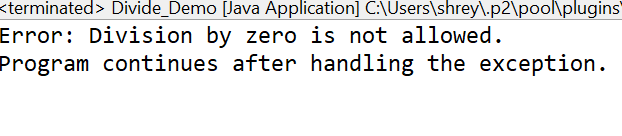
}

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.**

**Program**

**package** myPackage;

**import** java.util.Scanner;

**public** **class** Divide\_Demo {

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

// **TODO** Auto-generated method stub

**int** numerator = 10;

**int** denominator = 0;

**try** {

// Attempt to divide by zero

**int** result = numerator / denominator;

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

} **catch** (ArithmeticException e) {

// Handle the ArithmeticException

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

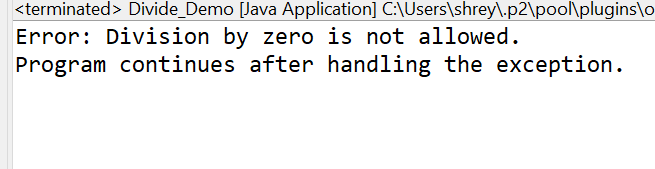
}

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

}

}

**Output:**



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

**Program**

**package** myPackage;

**public** **class** Arithitmatic\_Esception {

// Method to perform division and declare that it throws ArithmeticException

**public** **static** **void** divide(**int** numerator, **int** denominator) **throws** ArithmeticException {

// Perform division

**if** (denominator == 0) {

**throw** **new** ArithmeticException("Division by zero is not allowed.");

}

**int** result = numerator / denominator;

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

}

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

// **TODO** Auto-generated method stub

**int**[] numerators = {10, 20, 30};

**int**[] denominators = {2, 0, 8}; // Include a zero denominator for testing

**for** (**int** i = 0; i < numerators.length; i++) {

**try** {

System.***out***.println("Dividing " + numerators[i] + " by " + denominators[i]);

*divide*(numerators[i], denominators[i]);

} **catch** (ArithmeticException e) {

// Handle the ArithmeticException

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

}

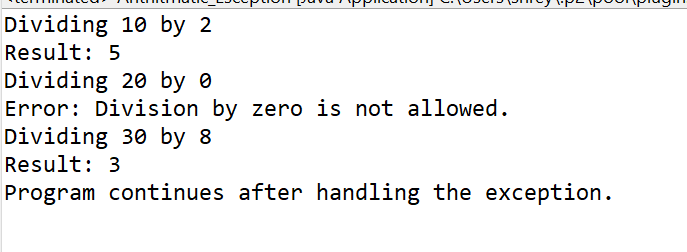
}

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

}

}

**Output**

****

**Q.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.**

**Program**

**package** myPackage;

**import** java.io.\*;

**public** **class** InvalidAge\_Esception **extends** Exception {

**public** InvalidAgeException(String message) {

**super**(message);

}

}

**public** **class** AgeValidationExample\_1 {

// Method to validate age

**public** **static** **void** validateAge(**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) {

// **TODO** Auto-generated method stub

**int**[] ages = {15, 20, 17, 22};

**for** (**int** age : ages) {

**try** {

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

*validateAge*(age);

} **catch** (InvalidAgeException e) {

// Handle the custom exception

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

}

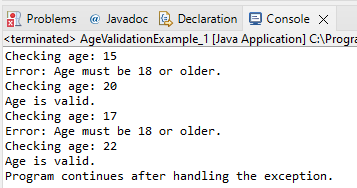
}

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

}

}

**Output**



**Q.6 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** myPackage;

**public** **class** InvalidEmail\_Exception **extends** Exception {

**public** InvalidEmailException(String message) {

**super**(message);

}

}

**public** **class** EmailValidationExample {

// Method to validate email address

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

// Check if email contains both '@' and '.'

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

**throw** **new** InvalidEmailException("Invalid email address. It must contain both '@' and '.'");

} **else** {

System.***out***.println("Email address is valid.");

}

}

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

// **TODO** Auto-generated method stub

// Example email addresses for testing

String[] emails = {"testexample.com", "user@domain", "user@domain.com", "valid.email@domain.com"};

**for** (String email : emails) {

**try** {

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

*validateEmail*(email);

} **catch** (InvalidEmailException e) {

// Handle the custom exception

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

}

}

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

}

}

**Output**

