|  |
| --- |
| **Department of Software Engineering**  **Mehran University of Engineering and Technology, Jamshoro** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Course: SWE121 – Object Oriented Programming** | | | |
| **Instructor** | Mr. Sajjad Ali | **Practical/Lab No.** | 09 |
| **Date** | 14-09-2021 | **CLOs** | CLO-3 |
| **Signature** |  | **Assessment Score** | 1 Marks |

|  |  |
| --- | --- |
| **Topic** | **Exception and Error Handling** |
| **Objectives** | * What is an Exception and how to handle exceptions in your program. * Learning the standard Java Exception provided by the Java API. * How to guarantee that a particular block of code will always be executed. * How to define and use the customized Exceptions. * How to throw Exceptions in your class. |

|  |
| --- |
| **Lab Discussion: Theoretical concepts and Procedural steps** |

**Tools:**  Java Development Kit, Text Pad, Netbeans, Eclipse

**Theory**

Outline

* Give an Overview of Exception Handling and Run-time errors
* To understand Exception Types
* Explain try….catch…..finally Blocks
* Write Java programs to implement error handling techniques using exception handling
* To lean Throwing Exceptions

**Exception Handling in Java**

The **Exception Handling in Java** is one of the powerful *mechanism to handle the runtime errors* so that normal flow of the application can be maintained.

In this page, we will learn about Java exceptions, its type and the difference between checked and unchecked exceptions.

**What is Exception in Java**

**Dictionary Meaning:** Exception is an abnormal condition.

In Java, an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

**What is Exception Handling**

Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc.

**Advantage of Exception Handling**

The core advantage of exception handling is **to maintain the normal flow of the application**. An exception normally disrupts the normal flow of the application that is why we use exception handling. Let's take a scenario:

1. statement 1;
2. statement 2;
3. statement 3;
4. statement 4;
5. statement 5;//exception occurs
6. statement 6;

**Types of Java Exceptions**

There are mainly two types of exceptions: checked and unchecked. Here, an error is considered as the unchecked exception. According to Oracle, there are three types of exceptions:

**Checked Exception**

**Unchecked Exception**

**Error**

**Difference between Checked and Unchecked Exceptions**

1) Checked Exception

The classes which directly inherit Throwable class except RuntimeException and Error are known as checked exceptions e.g. IOException, SQLException etc. Checked exceptions are checked at compile-time.

2) Unchecked Exception

The classes which inherit RuntimeException are known as unchecked exceptions e.g. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime.

3) Error

Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc.

Java Exception Keywords

There are 5 keywords which are used in handling exceptions in Java.

|  |  |
| --- | --- |
| Keyword | Description |
| try | The "try" keyword is used to specify a block where we should place exception code.  The try block must be followed by either catch or finally.  It means, we can't use try block alone. |
| catch | The "catch" block is used to handle the exception. It must be preceded by try block which  means we can't use catch block alone. It can be followed by finally block later. |
| finally | The "finally" block is used to execute the important code of the program.  It is executed whether an exception is handled or not. |
| throw | The "throw" keyword is used to throw an exception. |
| throws | The "throws" keyword is used to declare exceptions. It doesn't throw an exception.  It specifies that there may occur an exception in the method. It is always used with  method signature. |

|  |
| --- |
| **Lab Tasks** |

1. Write a java program in which exception is handled if number is divided by zero and print the exception name.
2. Write a java program in which takes an input from the user if his/her age is less than 18 then throw an user defined exception. User defined exception class must include two methods of eligible and not eligible.
3. Create a new InvalidPasswordFormatException which throws an exception on following scenarios:
4. Password is less than 8 characters.
5. Password has Asterik(\*)