**FACILITATOR’S MANUAL**

|  |
| --- |
| Facilitator’s manual is a guideline to facilitator. Guideline for which all topics /sub-topics to be covered and their sequence. When to go recap or hands-on and with which assignment (mapping of lab assignments with topics)  Basically WHAT–WHEN-HOW  Here, Whole session will be in multiple iteration of 3 steps;  1. What to facilitate, 2. Relevant LAB assignments, 3. Recap and leanings from LAB  Also, there are TIPS (extract from facilitator’s learning) – objective of TIPS is to incorporate best practice and individual’s innovation in facilitating a particular topic. It is desirable that new tips should continue to add/update in this manual.  At last, this is not a rulebook, so it is upto facilitator to follow it or use his/her own style |

**EXCEPTION & ASSERTION**

**Objective -** Understanding of exception and error, various handling mechanism, and exception hierarchy. Try-catch and finally block, throws, throw, checked and unchecked exceptions. To understand what is assertions and how to use it

**ROUND 1**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | **Exception**   * Concept of exception object * Exception flow (call stack) |
| LAB assignment | **LAB 8.1**  *Write a Class with main method; in main method divide a number with zero; run it and check behavior*  **LAB 8.2**  *Now write the divide by zero code in some method of different class, call the method from main method, run it. Check the exception stack on console.* |
| Recap (learning from the LAB assignment) | What is exception object and exception call stack |

**ROUND 2**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | **Exception**   * Exception handling mechanism * Try-catch-finally * Throws |
| LAB assignment | **LAB 8.3**  *Write a program, in which an exception occur; handle the checked exception with Try-catch*  **LAB 8.3 A**  *Modify 8.3 – use finally block as well and check how it works*  **LAB 8.4**  *Handle the exception of LAB 8.1 with throws* |
| Recap (learning from the LAB assignment) | Handle exception using try-catch-finally block  Handle exception using throws |

**ROUND 3**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | **Exception**   * Nested try-catch |
| LAB assignment | **LAB 8.5**  *Write a program, where class has many possibility of occurrence of an exception. Handle them using nested try catch block.* |
| Recap (learning from the LAB assignment) | Nested try catch |

**ROUND 4**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | **Exception**   * Throwable class hierarchy * Error and various type of exception * Checked and unchecked (runtime) exception |
| LAB assignment | **LAB 8.6**  *Write a program with a checked exception code; do not handle it and try to run the program.*  **LAB 8.7**  *Write a program, where class cls1 is calling a method (divide two numbers) of another class cls2, while calling this send two number in such a way that method generate divide-By-Zero or arithmetic exception;*  *now run the following five aspect -*  *1. do not handle exception at any place*  *2. handle it at cls2 method using throws*  *3. handle it at cls2 method using try catch and do not handle at calling class*  *4.handle it at cls2 method using throws and try catch at calling method of cls1*  *5.handle it using try catch at both cls2 and calling class*  *This exercise to understand the exception handling stack and how it traverses* |
| Recap (learning from the LAB assignment) | Difference between checked and unchecked exceptions  Checked exception are compulsory to handle |

**ROUND 5**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | **Exception**   * Throw * User define exception, how to use it |
| LAB assignment | **LAB 8.8**  *Write a user-define exception, and throw it from some other class and handle it. Aim of this assignment is to understand that whether user defined exceptions are checked or unchecked and how do we handle them.* |
| Recap (learning from the LAB assignment) | How to write our own exception  How to handle user defined exception  User define exception (which extends Exception class) is checked expection |

**ROUND 6**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | **Assertions**   * Introduction of assertion * How to use it * Two types of assert expression * Enabling assertion while running java code |
| LAB assignment | **LAB 8.9**  *Write a program to check simple assertion value. Run this program when assertion is disabled and when it is enabled.*  **LAB 8.9 A**  *Modify 8.9 - use second style of writing assertion.* |
| Recap (learning from the LAB assignment) | How and why to use assertion |