**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 |

**MULTITHREADING PROGRAMMING IN JAVA**

**Objective -**  To understand, concept of multithreading programming, how java support it, advantages and drawback of multithreading programming. To know about various ways of thread safety, inter thread communication and various methods of thread.

**ROUND 1**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Thread overview * Difference between thread and process * Thread Life cycle (States of thread) * How to create runnable (thread) object,  1. Implementing Runnable 2. Extending Thread  * Sleep() method * Various constructor of thread |
| LAB assignment | **LAB 10.1**  *write a program where, class cls2 will have a method display – which prints table of 2 using for loop; now from another class cl1 (having main method) create two object s of cls2 and call display method.*  *Now convert cls2 as a runnable one, using runnable implementation. Override run method, and call display from run method.*  *Now the constructor of cls2 should create two thread objects. And from cls1 now create only one object of cls2 and don’t call display method (as the display method will now be called from run method)*  *See the different in output in both the cases – you will find that in first scenario table of 2 was printing 2 times, one by one… using thread the table is printing simultaneously - Aim of this assignment is to see behave of thread, benefit of thread (as now the same code can be execute simultaneously by various thread) , how to use runnable interface to create thread object*  **LAB 10.1 A**  *Modify 10.1 now, also give names to the thread, and display them in result so that, we can have idea of which thread is executing when.*  **LAB 10.2**  *Modify Lab 10.1 A; this time use Thread class (in placeof Runnable interface) to make objetc of Cls2 runnable.* |
| Recap (learning from the LAB assignment) | Understanding of thread, and multithread execution  Two ways of creating runnable object  Various constructor of Thread |

**ROUND 2**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Thread priority * Various methods of thread (getName, setName, getProirtity, setPriotity, isAlive, currentThread,etc) |
| LAB assignment | **LAB 10.3**  *Create a runnable class, and run multiple threads (We already did so in Lab 10.1 and 10.2) now use various methods of thread* |
| Recap (learning from the LAB assignment) | Understaning of various methos of thread like getName, setName, getProirtity, setPriotity, isAlive, currentThread,etc. |

**ROUND 3**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Thread safety concept * Synchronization technique * Concept of lock |
| LAB assignment | **LAB 10.4**  *Write a stack-class, which should have Pop and push method and check if stack is full or empty prior to pushing or popping, so that out of array exception can never occur using this stack.*  *Now write a program, where one thread is pushing the objects in stack while other is popping it out, run the two threads simultaneously, so that array-out-of-bound exception occur; which means common resource (stack) is being used by two thread and chances of resource manipulation has occurred.*  *Aim of this assignment is to create thready safety issue.*  **LAB 10.5**  *Make the program of LAB 10.4 as thread safe with the help of synchronization.* |
| Recap (learning from the LAB assignment) | Understaning of thready safety concern  Synchronization and various synchronization techniques |

**ROUND 4**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | Inter Thread Communication   * Various method of thread intercommunication like wait(), notify(), yield() etc. * Dead Lock * Deamon thread * Timer class |
| LAB assignment | **LAB 10.6**  *Write a program, and use various methods like wait(), notifyAll(), yeild(), etc.*  **LAB 10.7**  *Write a program, which creates a dead lock situation.* |
| Recap (learning from the LAB assignment) | How and when to use various methods for thread intercommunication  What is dead lock and how to avoid it |