**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 |

**Topic - Object and Classes**

**Objective -**  To understand Classes, static and non-static member (variable and method), object instantiation, calling a method, passing arguments to a method, sub class and super class, method overriding and overloading.

**ROUND 1**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Recap on what is object and class * Create/instantiate an object * Method and Method signature * Constructors * Constructors overloading * Calling of a method |
| LAB assignment | **Refer - LAB 3.1**  *Create two java classes, now in class1 (which will have main method) create object of the class2 (which will have a method called display () - which displays a message on console). Now call the display method of Class2.*  **Refer - LAB 3.2**  *Now override the default constructor of class2 (cls2), display some message within constructor. This exercise is to understand default constructor.*  **Refer - LAB 3.3**  *Now add a parameterized constructor of Cls2, and instantiate another object of Cls2 in cls1 using this.*  *Now remove the default constructor of Cls2, and try to run the code.* |
| Recap (learning from the LAB assignment) | How to create an instance/object of a class  Knowledge of Constructor of a class  Parameterized constructors  How to call a method of a class |

\*TIPS TIME – while creating workspace and java project, ask associate/participant to check the same in their system, if the files and folder are automatically creating in the respective (defined location) in their system.

**ROUND 2**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Passing values as an arguments to a method and accepting return value |
| LAB assignment | **LAB 3.4**  *Create two java classes -, first, Mainclass (which will have main method) and second, MyMath Class having methods like add, subtract, multiply, divide (each methods will have two parameters, that is will perform respective mathematical operation between two numbers). Now call the various methods of MyMath class from MainClass and pass the two numbers, receive results and print them.*  **LAB 3.5**  *In Lab 3.4; receive the two numbers for user as runtime argument, now use these numbers.*  **LAB 3.6**  *Further improvement (variance in Lab 3.5) receive the two numbers and type of operation from user as runtime argument (command line) and use either if-else-if or switch block to call appropriate method of MyMath class.* |
| Recap (learning from the LAB assignment) | How to pass the values to a method  What is return type of a method  How to receive the value/info returned from the called method |

*\*Lab 3.5 and 3.6 is optional; it is just to create further interest among participants*

**ROUND 3**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Static and non-static members (variable and method) |
| LAB assignment | **LAB 3.7**  *A class will have a global variable (say int x); now create the two objects of this class in main class (having main method), now change the variable value in both the object (value should be diff, say object1.x = 10 and object2.x = 12), print this value.*  *Now make the global variable static and run the same program. Analyze the different in result.*  **LAB 3.8**  *This is extension of Lab 3.7; now write two methods in class, one static and one non static, now create object of this class in main method, call this methods; call the method without using object instance (directly from name of class).*  **LAB 3.9**  *Write a java class having main method; now define some non static and some static global variable and try to access them in main method.* |
| Recap (learning from the LAB assignment) | Understand difference between static and non-static variables  Understanding that static method (members) of a class can be accessed without object instance of that class.  Static method cannot access non-static members. |

**ROUND 4**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Over loading |
| LAB assignment | **LAB 3.10**  *At present all methods of MyMath class (Add, Subtract, Multiply, Divide) have two parameters (i.e. for two numbers only). Now the business requirement is to have all these method for 3 parameters as well.*  *Overload all methods for 3 parameters. (Do not modify old methods, add new methods with same name and return type with different number of parameters)*  *Additionally, check if overloading is applicable*   * *In case of same number of parameters but for different data type.* * *In case of different return type but same parameters* |
| Recap (learning from the LAB assignment) | What is method overloading, how and why |

**ROUND 5**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Sub Class * Super Constructor * Overriding * Difference between overriding and overloading |
| LAB assignment | **LAB 3.11**  *Write a java class - MySubMath which extends the MyMath class; Create an instance (object) of MySubMath class in MainClass (class having main method), and check if all methods of MyMath is available with the instance of MySubMethod.*  **LAB 3.12**  *Override all/some methods of MyMath in MySubMath (like add 5 in the result and then return it, this is just to distinguish the method operations in Super and Sub (overridden) class.*  *Create instance of both the classes in MainClass (in main method) and call the same methods of both the class with same argument values. (Example – call add method from both MyMath and overridden add method from MySubMath class). Check the result.*    *Try to extends more than one class (i.e. try to create many super class of a sub class)* |
| Recap (learning from the LAB assignment) | How to create a sub class  A subclass by default inherits all members of super class.  A sub class can have at most only one super class  Understanding of overriding (what, how and why)  Difference between overriding and overloading |

**ROUND 6**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * This constructor * Super Constructor |
| LAB assignment | **Refer - LAB 3.18**  *Practice on This*  **Refer - LAB 3.19**  *Practice on Super* |
| Recap (learning from the LAB assignment) | This  Super  Should always be first |

**ROUND 7**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Final Class * Final modifier for method and properties |
| LAB assignment | **LAB 3.13**  *Write a final class; and try to extends this class*  **LAB 3.14**  *Write a normal public class, but write a final method inside it; now try to override it* |
| Recap (learning from the LAB assignment) | What is final class  How to create a final class  Final classes cannot be extends (thus modified or override)  Final Method |

**ROUND 8**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Call by reference and call by value |
| LAB assignment | **LAB 3.15**  *Write a program; where class1 will call 2 methods of class2, first method will take primitive data as argument and second will take object of class one. Now change the primitive data value in method 1 for class 2 and check if this value is getting changed in class1; check the similar situation for method 2.*  *Aim of this program is to understand call by value and call by ref* |
| Recap (learning from the LAB assignment) | Difference between call by value and call by reference (object ref) |

**ROUND 9**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Inner class concept * Why we need inner class * HAS – A, IS-A relationship * Static and non static inner class * Local class and Annonimous inner class |
| LAB assignment | **LAB 3.16**  *Write a program; where class Outer will have a inner class; both outer and inner class will have a display ( ) method, which display a message.*  *Now create object of inner class and call its display method from some other class.*  **LAB 3.17**  *Modift Lab 3.15, add a static inner class, which will have a display () method as well. Now call this method from some other class.* |
| Recap (learning from the LAB assignment) | Revisited all class related concepts, except for access and other modifier, which we will cover after “packages” will be covered |

**ROUND 10**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Recap - Member of a Class – Local/global variable, constructor, method, method signature, static and non static |
| LAB assignment | N/A |
| Recap (learning from the LAB assignment) | Revisited all class related concepts, except for access and other modifier, which we will cover after “packages” will be covered |