**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.  x  At last, this is not a rulebook, so it is upto facilitator to follow it or use his/her own style |

**CORE-JAVA LANGUAGE BASIC**

**Objective -**  To understand basic concepts like – operator, variable, constants, identifier, etc. and syntax for them.

**ROUND 1**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Identifier * Reserved key words \* * Constrains for Identifier   + Can start only with $, \_ or alphabets   + Can have numbers   + Can’t user reserved key words * Data type   + byte(8), short(16), int(32), long(64); float(32), double (64); char (16); boolean * Variables (Global and Local) * Constants |
| LAB assignment | **Refer - LAB 2.1**  *Write java program (Class) in which define some global of different data type (int, short, byte, char, boolean, etc.) and then print their default values.*  **Refer - LAB 2.2**  *Modify LAB 2.1; now initialize the global variables and print them and then modify them and again print them.*  *Now also create some constants; and try to modify them as well.* |
| Recap (learning from the LAB assignment) | LAB 2.1   * How to define a global variable * Use of data type in creating a variable * Default value of global variables * Reserve keywords cannot be use as identifier   LAB 2.2   * How to define a constants * Global variables can be modify in a method (but will be locally applicable) – will see it in future examples |

*\*****TIPS TIME*** *– To make reserve key word session more attractive – you can divide the class into two groups, and one by one each group has to tell a valid reserve keywords; the group which tells more number for reserve keyword will win.*

*This exercise helps to remember it as well as create a lighter environment in class. You will surprise to find that even without prior reading for reserve keywords in java, class will be able to tell most of them based on their aptitude.*

**ROUND 2**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Data type casting   + Byte, short, char, int, long, flaot, double   + Up-casting and down-casting |
| LAB assignment | **Refer - LAB 2.3**  *Write java program (Class) in which – define some variable of different type say int, shor, byte, long, etc. data type; now try to cast (assign) the int variable to a short variable, and vice versa; int value to long and vice versa.*  *Basically try various permutation and combination for up& down castings.* |
| Recap (learning from the LAB assignment) | Understanding of various data type casting  Up casting (explicit) and Down casting (implicit)  Casting leader – byte 🡪 short 🡪char 🡪 int 🡪 long 🡪 float 🡪 double |

**ROUND 3**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Operator   + Unary (++ --) binary (arithmetic, relational, logical, bitwise (logical, shift), ternary * Operator precedence   + Casting, bracket, Unary ++ --, arithmetic (\* …. +), relational (> <…..== !=) logical (!, &&, ||) |
| LAB assignment | **Refer - LAB 2.4**  *Write java program (Class) in which – define some numeric variables (int, short, etc.) perform some arithmetic, shift and unary operations. Check the precedence’s. \**  **Refer - LAB 2.5**  *Write java program (Class) which take command line arguments and perform some mathematical operation on them.* |
| Recap (learning from the LAB assignment) | Understanding of various operations, operator and operator precedence |

*\*****TIPS TIME*** *– Ask the class to practice it as much as possible, ask class/participant to think of scenario by them self and check their activity. Discuss the individuals learning with class.*

**CORE-JAVA LANGUAGE BASIC (Flow Control)**

**Objective -**  To understand various flow control and looping mechanism in Java like – If-else, switch, for loop, while loop, do-while loop, break, continue, etc.

**ROUND 4 - If else**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * If; * If-else; * if-else-if |
| LAB assignment | **Refer - LAB 2.6**  *Write java program (Class) which take 3 command line arguments (numbers) and perform some mathematical operation on last two numbers, first number will be use to tell what operation user want to perform (1 for (+), 2 for (-), 3 for (\*), 4 for (/) and 5 for (%))* |
| Recap (learning from the LAB assignment) | Use of if block, if-else bock, if-else-if block  Use of logical and conditional expression (operations) |

**ROUND 5 - SWITCH**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | **Switch**   * Switch accept byte, int, short, char and enum type as argument; from java 7, it is accepting String * Switch is helpful only if validating condition is equal to (not applicable for logical condition) * Break is necessary to write |
| LAB assignment | Refer - LAB 2.7  *Rewrite LAB 2.6; this time in place of if-else-if use switch block.*  *Now, again use LAB 2.7, this time just remove break statement from switch block.* |
| Recap (learning from the LAB assignment) | How to use switch block  Important of break statement  Valid argument (in terms of data type) for switch condition |

**ROUND 6- flow control – FOR**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | **FOR**   * Normal for loop * Infinite for loop - for ( ; ; ) |
| LAB assignment | **Refer - LAB 2.8**  *Write table of 3 with the help of for loop (display every number in a new line)*  **Refer - LAB 2.9**  *Write table of 3 with the help of for loop; but every third no. (which is 9, 18, 27 in table of 3), should not be visible* |
| Recap (learning from the LAB assignment) |  |

**ROUND 7**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | **FOR**   * Break * Continue * Nested loop |
| LAB assignment | **Refer - LAB 2.10**  *Write table of 3 with the help of for loop (display every number in a new line)*  *; On every third no. (which is 9, 18, 27 in table of 3),this no should not be visible rather there will be table of 10, 20, 30 (display in single line) respectively* |
| Recap (learning from the LAB assignment) |  |

**ROUND 8: while and do-while**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * While * Do-while |
| LAB assignment | Rewrite 2.8, 2.9 and 2.10 with help of while and do-while loop |
| Recap (learning from the LAB assignment) |  |

**CORE-JAVA LANGUAGE BASIC (ARRAY)**

**Objective -**  To understand what is array is Java; use of array, how to create and initialize an array, multidimensional array.

**ROUND 9 - Array**

|  |  |
| --- | --- |
| Topics to be facilitated (teach) | * Array * Creation and initialization of array * Multidimensional array * Drawbacks of an array |
| LAB assignment | **Refer - LAB 2.12**  *Write java program (Class), in which, we will create an array, initialize, populate and use it.*  **Refer - LAB 2.13**  *Now modify the LAB 2.12, try to enter more elements in array then its defined size; OR try to run the loop, which prints value of array for more times than number of array elements. Purpose of this exercise is to check limitations of array.*  **Refer - LAB 2.14**  *Write java program (Class), in which, create a two dimensional array, initialize it, and populate with value of table of 2.* |
| Recap (learning from the LAB assignment) | Use and understanding of array  Understanding of its limitations |

Post the session on Classes and Object, we will do more complicated exercise (assignments) on array.