Java Syllabus: -

Core Java Masters Batch - Month 1

I. Fundamentals and Primer of Java

a. Java Primer

i. Getting Started

(I). What is a Programming Language.

(II). Why to use Programming languages.

(III). What is Java.

(IV). Why to choose Java.

(V). History and Future of Java.

(VI). Basics of Variables, Datatypes and Literals.

(VII). Writing our First Java Program.

(VIII). Compilation Unit of Java.

(IX). Command Line Arguments and JVM.

(X). JDK, JRE and JVM.

(XI). Summary of Current Section.

ii. Identifiers and Variables

(I). What are Identifiers?

(II). Rules of naming Identifiers.

(III). Understanding more about Datatypes.

(IV). Creating Variables.

(V). Using Variables and Datatypes Concept to find Sum of 2 Numbers.

(VI). Using Operators.

(VII). Test your Knowledge on OCJP Questions.

(VIII). Summary of the Section.

iii. Datatypes

(I). What are Datatypes.

(II). Segregation of Datatypes.

(III). Primitive Datatypes

a. byte

b. short

c. int

d. long

e. float

f. double

g. boolean

h. char

(IV). Reference Datatypes

a. Understanding briefly about Strings.

(V). Revisit to Variables and Identifiers.

(VI). Literals or Constants

(VII). Type Casting

a. Primitive

b. Reference (Basic Level).

c. Implicit and Explicit Casts

d. Common Errors in Type Promotion/Demotion.

(VIII). Local Variable Type Inferencing.

a. Using context-specific identifier var.

b. var restrictions.

c. Updates of var.

iv. Control Flow in Java Program.

(I). Types of Statements.

(II). Sequencial statements.

a. What, where and when to use these statements.

(III). Selective statements.

a. All variants of if-else

b. switch-case

c. Fall through

d. Understanding what construct to be used according to a program question.

(IV). Iterative statements.

a. for()
b. Variations of for() loop.
c. for-each
d. while()
e. do-while()
f. When and where to use the constructs wisely.
g. Jump Statements.
(i). break
(ii). continue

v. Math operations in Java
(I). pow()
(II). sqrt()
(III). floor()
(IV). ceil()
(V). round()
(VI). rint()
(VII). round() vs rint()
(VIII). Using Math methods wisely.
(IX). Solving OCJP Questions from Enthuware.
(X). Summary

2. Operators
(I). Arithmetic Operators
(II). Bitwise Operators
(III). Relational Operators
(IV). Logical Operators
(V). Short Circuit Operators
(VI). Assignment Operators
(VII). Ternary Operators.
(VIII). Operator Precedence.
(IX). Parentheses.
(X). Expressions

3. Arrays

(I). Introducing Arrays Concept.

(II). Array Initializer

(III). Anonymous Arrays

(IV). One Dimensional Arrays

(V). ID Array Declaration, Creation and Initialization.

(VI). Accessing Elements in ID Arrays.

(VII). 2D Arrays

(VIII). Multi-Dimensional Arrays

(IX). Jagged/Ragged Arrays.

4. Strings, StringBuffer and StringBuilder

(I). Introduction.

(II). Important Constructors of String class.

(III). All methods and Variants of String class.

(IV). Important Conclusions of String class.

(V). Getting started with StringBuffer.

(VI). Understanding StringBuilder.

(VII). StringBuffer vs StringBuilder.

(VIII). Thread-Safe

5. Classes and Objects

(I). Class Fundamentals

(II). Declaring objects.

a. new keyword

(III). Working of Objects and classes internally.

(IV). Introducing Methods.

(V). Constructors

(VI). this keyword

(VII). Garbage Collectors in Java

(VIII). class design

(IX). Advanced Class Design.

6. Closer Look at Methods and Classes.

(I). Overloading Methods.

(II). Using Objects as Parameters.

(III). Argument Passing.

(IV). Call by value and Call by Reference.

(V). Returning Objects.

(VI). Recursion

(VII). Access Control

(VIII). Using static keyword.

(IX). final keyword in Java.

(X). Nested classes.

(XI). Var-args.

(XII). Ambiguities in var-args

(XIII). Local Variable Type Inferencing in Reference Types.

7. Inheritance

(I). Basic Introduction.

(II). Using super keyword

(III). Multilevel Hierarchy

(IV). Execution of Constructor.

(V). Method Overriding

(VI). DMD

a. Overriding Methods

b. using method overriding.

(VII). Abstract classes.

(VIII). Using final with Inheritance.

(IX). Local Variable Type Inferencing with Inheritance

(X). Object class.

8. Packages and Interfaces

(I). Introduction

(II). Representation of Packages

(III). Usage of Packages.

(IV). Revising Naming Conventions of Packages.

(V). Questions of OCAJP on Packages.

(VI). Interfaces Introduction

(VII). Using abstract and strictfp modifiers.

Bonus : Learning about Default Methods and updates of JDK 8

(VIII). Initialization of Fields in Interfaces

(IX). Super and Sub Interfaces

(X). Interface Members

(XI). Inheritance, Overriding and Overloading in Interfaces

(XII). Functional Interfaces

9. Exception Handling

(I). Introduction

(II). Best Example to Understand Exception Handling.

(III). Types of Exceptions

(IV). Differenciation between Checked and Unchecked Exceptions

(V). More on Unchecked Exceptions

(VI). try and catch

(VII). Stack Trace

(VIII). Understanding Default Exception Handler and Method Stacks.

(IX). Multiple catch clauses and restrictions

(X). Nested try statements.

(XI). Creating our own Exceptions

(XII). throw keyword

(XIII). Throwing a Checked Exception

(XIV). finally block

(XV). Creating a Project which uses its own Exceptions.

(XVI). Java's Built-In Exceptions.

(XVII). StackTraceElement class

(XVIII).Updates in Exception handling in Java with subsequent releases.

10. Multithreaded Programming

(I). Introduction to Multithreading Concepts.

(II). Thread Model

(III). Types of Multithreading.

(IV). Round-Robin Model

(V). Synchronization

(VI). Two ways of creating our own Thread subclasses.

(VII). Thread class and Runnable Interface

(VIII). Main Thread

(IX). Creating a Thread

(X). Creating Multiple Threads

(XI). Overriding run() method.

(XII). Synchronization in Depth

(XIII). Inter-Thread Communication

(XIV). Suspending, Resuming and Stopping Threads.

(XV). State of a Thread.

(XVI). Applications of Multithreading.

(XVII). Banking Project of Multithreading.

11. Enumerations, Annotations and Autoboxing.

(I). Enumerations Fundamentals.

(II). values() and valueOf()

(III). Enums and Enumerations

(IV). Example

(V). Type Wrappers

a. Character

b. Boolean

c. Numeric Wrappers

(VI). Autoboxing

(VII). Methods of Autoboxing

(VIII). Autoboxing in Expressions

(IX). Unboxing

(X). Preventing Errors.

(XI). Usage of Annotations

(XII). Few Annotations

(XIII). Summary

12. Handling User Inputs, I/O and Console

(I). Streams Introduction

(II). Reading Console Inputs

(III). Console Outputs

(IV). PrintWriter class

(V). Reading and Writing a File using Scanner class.

(VI). Reading and Writing a File using BufferedReader class.

(VII). A bit about NIO --Java 7

13. Generics

14. Lambdas and Functional Interfaces.

(I). Fundamentals

(II). Functional Interfaces Revision

(III). Block Lambdas

(IV). Generic Functional Interfaces

(V). Passing Lambdas as Arguments.

(VI). Using all types of Functional Interfaces.

a. Predicate

b. BiPredicate

c. Consumer

d. Function

e. BiConsumer

(VII). Summary

15. SELF EVALUATION PRACTICE ON OCAJP QUESTIONS.