**JAVA**

**“If you are computer engineer, you probably think of java as a computing platform rather than as a coffee or an island in the pacific”**

**First, though, you need to understand why should you choose java as career even there are lots of programming language in the world…**

* **“java as a celebrity”: According to the TIOBE index for February 2016 and IEEE spectrum ranking, java was crowned as the most popular computing language in the world.**
* **“A bank book makes good reading better than some novels”** 
  + - **According to <gooroo_report.html> : java holds the title as most paying language in world.**
    - **As per Indeed.com, average salary of java developer is more than $100,000.**
* **“Java is everywhere”: As per oracle java is estimated to be running over 3 billion devices all over world. No other language runs on as many devices.**
* **There are so many developer tools and third party api’s provided to support java.**
* **Java is used in many real world application. It includes various prominent web application, android app, cloud computing and data analytics etc.**
* **Worldwide there are around 9 million java developers, java has a very active and passionate community across the world.**
* **As java being so much popular, java learning material is easily available over worldwide on internet. That’s make java very much easy to learn and to work as well.**
* **As a beginner, we can easily learn java because there is no such prerequisite like any basic of programming, but having any basic idea of any programming language will be act like a catalyst in the learning process.**
* **One of the most Well respected certification in java by Oracle Inc.**

**1. Introduction to advance programming**

i) Difference between basic programming and advance programming

ii) History of java

iii) History of python

iv) Java versus Python. (In the aspect of technical terms).

v) Market survey of both the languages

vi) Concept behind object oriented programming paradigm

vii) Research paper related topics discussion

viii) Final project related topic discussion

ix) Guideline and timeline for research paper and final projects

x) Conclusion and summary

**2. Introduction to java**

i) Concept behind java

ii) Java as a high language

iii) Complier versus interpreter

iv) Introduce with JDK.

v) Comparing all versions of JDK.

vi) Familiar with java run time environment

vii) Java download and installation

viii) Setting up the environment

ix) Writing the first java program

x) Saving the source file, compiling and run your first java program and get the output

**3. Brief discussion on internal functionality of JDK.**

i) In depth discussion on first program syntax

ii) Introduction to java complier

iii) Java .class file and magic code

iv) Implicit java class loading

v) Various class loaders

vi) Explicit class loading in java

vii) Java interpreter working

viii) Java performance discussion

ix) Introduction to JIT environment

x) Brief discussion on memory management in java

**4. Fundamental programming structures in java**

i) A simple java program

ii) Comments

iii) Data types

iv) Variables

v) Operators

vi) Introduction to arrays

vii) String fundamental

viii) Conditional statements syntax

ix) Looping in java

x) Memory allocation of different data types

**5. Introduction to object oriented programming paradigm**

i) Oops as a methodology

ii) Object oriented programming versus classical programming paradigm

iii) What is an object

iv) Understanding of class

v) Different type of fields in a class

vi) Association

a) Aggregation

b) Composition

vii) Polymorphism

viii) Abstraction

ix) Encapsulation

x) Inheritance

**6. Implementation of OOPS**

i) Object

a) Different ways to create an Object

 Static Factory Method

 Constructor

b) Comparing all the ways to create an object

 Static Factory Method Vs Constructor

c) Self-executing blocks

 Init Block

 Static Block

d) Introduction to this

ii) Class

a) Defining your own class

b) Class Scope

 Introduction to Packages

 Public

 Protected

 Default

 Private

iii) Implementation Of Association

a) Aggregation

iv) Implementation of Polymorphism

a) Method Overloading

b) Constructor Overloading

c) Constructor chaining

v) Implementation of Inheritance

a) Introduction to extend keyword

b) Introduction of super Keyword

c) Dynamic Polymorphism Implementation

vi) Implementation of Abstraction

a) Abstract Class

b) Abstract Method

c) Interfaces

vii) Implementation of Encapsulation

viii) Introduction to final Keyword

a) Class

b) Variables

c) Methods

ix) Implementation of composition

a) Introduction to nested classes and interfaces

 Static nested class

 Inner class

 Anonymous class

 Local class

Assignment**..**

**7. Methods common to all**

i) General contract when overriding equals()

ii) hashCode()

iii) clone()

iv) compareTo()

v) toString()

vi) == operator

vii) Introduction to wait(), notify() and notifyAll()

viii) Finalize()

ix) Introduction to garbage collector

x) Call of System.gc()

**8. Introduction to arrays**

i) Definition

ii) Types

a) One Dimension

b) Two dimension

c) Multi dimension

iii) Declaration

iv) Initialization

v) Resizing

vi) Accessing

vii) Copying

viii) Searching

ix) Sorting

x) Introduction to java.lang.Arrays

**9. Strings**

i) Introduction of string family

a) Java.lang.String

b) Java.lang.StringBuffer

c) Java.lang.StringBuilder

d) Java.util.StringTokenizer

ii) String creation

a) String literals

b) Via new operator

c) Comparison between both

iii) Concept of memory management of string

iv) Concept of immutable

a) What is immutable?

b) Why string is immutable?

c) How to create your own immutable object

d) Use of immutable object

v) String comparison

a) equals()

b) == operator

c) compareTo()

d) equalsIgnoreCase()

vi) toString() implementation

vii) String manipulation

a) concat()

b) + operator

c) Performance comparison between both

viii) Introduction to inbuilt methods of String class

ix) Comparison among String, StringBuffer and StringBuilder

x) Implementation of StringTokenizer

**Assignment..**

**10. Exception Handling**

i) Dealing with errors

ii) Concept of exceptions

iii) Difference between exceptions and errors

iv) Types of exceptions

a) Checked

b) Unchecked

v) Introduction to Throwable class

vi) try, catch and finally blocks

vii) Protocols for implementing different blocks

viii) Concept of throw and throws

ix) Introduction to customs exceptions

x) Use and implementation of own exceptions

**11. Play with the threads**

i. Introduction to Program, Process, Thread

ii. Definition of:

a) Multitasking

b) Multiprogramming

c) Multiprocessing

d) Multithreading

iii. Concept of Scheduler

iv. Thread Vs Process

v. Process Scheduling

vi. Creation of thread using java api’s

a) Java.lang.Thread

b) Java.lang.Runnable

vii. Thread state model

a) Implementation of each and every state

viii. Thread class methods

a) Implantation of all the methods

ix. Thread interruption

x. Concept of thread synchronization

a) Synchronized block

b) Synchronized method

xi. Liveness

a) Deadlock

b) Starvation and Livelock

xii. Inter thread communication

a) Implementation of wait() and notify()

xiii. Introduction to Shutdown hook

**Assignment..**

**12. Introduction to concurrency**

i) Introduction to concurrency

ii) High level concurrency object

 Lock object

 Executors

iii) Executor

iv) Thread factory

v) Futures

vi) Queues

vii) Synchronizers

viii) Atomic

ix) Locks

x) Applications

xi) Concurrent hash maps

**13. File I/O**

i) An overview of the java.io package

ii) Java.io class overview

iii) Sources and sinks

iv) File

v) Buffering

vi) Filtering

vii) Data I/O

viii) Object serialization

ix) Tokenizing

**14. Networking**

i) Introduction

ii) Important classes

iii) Connection Oriented communication with ServerSocket and Socket classes

iv) The Socket class

v) The ServerSocket class

vi) What is connectionless communication

vii) The DatagramSocket Class

viii) Exceptions

**15. RMI**

i) The RMI feature

ii) Object Persistence and RMI

iii) The RMI Architecture

iv) The steps for building RMI application

**16. Collections and Generics**

i) What Do You Do with a Collection?

ii) Key interfaces and Classes of the Collections framework

a) Collection interface

b) Iterator interface

c) Set interface

d) List interface

e) Map interface

iii) Sorting

iv) Thread-safe collections

v) Historical collection classes

a) Vector

b) Enumeration interface

vi) Generics in the Java Programming Language

vii) Introduction

a) Defining Simple Generics

b) Generics and Subtyping

c) Wildcards

d) Bounded Wildcards

**17. JDBC**

i) Introduction to JDBC

ii) JDBC Product Components

iii) Describe JDBC Architecture

iv) Explain types of Driver

v) Create JDBC Application

a) To load the Load Driver

b) To create Connection object

vi) To create

a) Statement

b) PreparedStatemnet

c) Callable object

vii) Types of ResultSet

viii) To execute Query

ix) To traverse ResultSet object

x) To close the statement

xi) To close Connection.

**\*. Logging, Assertion, Debugging**

i) Assertion

a) Introduction to Assertion

b) Using assertion

c) Enabling and disabling

d) Using assertion for parameter checking

e) Using assertion for documenting assumptions

ii) Logging

a) Introduction to logging

b) Basic

c) Advance

d) Log manager configuration

e) Localization

f) Handler

g) Filters

h) Formatters

iii) Debugging

a) Debugging tips

b) How to use debugger

iv) Linux

v) Basic SQL