**JAVA(Day -1)**

* **Core Java:**

**Fundamentals of Java:**

**OOPS:**

**4 Pillars and class and objects**

**Data Types:**

**Exception handling:**

**MultiThreading:**

**IO:**

**Collections:**

**JDBC:**

**Features 1.8, 9, 10, 11:**

**======================================================>**

**HTML CSS**

**======================================================>**

**Advanced Java:**

**Servlet**

**JSP(EL)**

**======================================================>**

**Frameworks:**

1. **Hibernate(JPA)**
2. **Spring**
3. **SpringBoot**

**======================================================>**

**History:**

* **1995- James Gosling and his team**
* **First Name - OAK**

1. **OAK Tree as a symbol of strength**

* **Second Name - JAVA**

1. **Java was put name due to team effort as such saying, Java was a brand name for coffee and everybody was liking it**

**Program:**

**A set of instructions**

**Process:**

**An execution of a program**

**=======================================================================**

**Programming Types:**

1. **Procedural: C (Everything is written in main function)**
2. **Object - oriented:**
3. **Partially: no tight dependency on objects. Everything is not related to object. CPP, JAVA**
4. **Fully : Tight dependency on objects. SCALA**
5. **Object-based : DOM, VB SCRIPT, JAVASCRIPT**

**=======================================================================**

**Levels of Programming:**

1. **LOW : It is a binary language. Technically Expert is mandatory.**
2. **MODULARITY : LOW + HIGH. Pointers, Storage classes + functions(modularity), for**
3. **HIGH : Human Readable. English Language.**

**=======================================================================**

**Statement = Java is high Level Language which is partially oops.**

**=======================================================================**

**Features :**

1. **Simple: Pre-requisites is CPP.**
2. **Architecture Neutral: not specific to architecture. The size of primitives is same for all architecture**
3. **Platform Independent: Write once run anywhere**
4. **Secure: no usage of pointers. No memory leakage. Garbage collection is automatic. Exceptions class(Java contains predefined exception class).**
5. **Interpreted**
6. **Robust**
7. **MultiThreading**
8. **Dynamic: var can also be used in java.**
9. **Distributed: after building API, can share anywhere.**

**SONARLINT**

**Process -> Program(smallest unit of process) -> Thread(smallest unit of program)**

**Default value of character is unicode(‘\u0000’)**

**JDK, JRE, JVM**

**JIT**

1. **JDK is platform independent**

**Who is platform dependent and independent ?**

1. **Can we explicitly download jre ?**