Day3 OOPs concepts in JAVA.

1. Class
2. Object
3. OOPs vs POPs
4. Abstraction
5. Inheritance
6. Encapsulation
7. Polymorphism

Day2 Revisit

Primitive & Non-primitive data types

Class vs Object

Eclipse & Intellij

Class – Blueprint [user – name, email, dob, mobile, etc.,]

Class will have state and behavior (properties & methods)

A class can have n no of properties and methods ().

Methods – It’s function defined inside class.

Display(), show(), calculateInterest()

Types of methods

1. With Arguments with return
2. With Arg no return (void)
3. No Arg with return (int, float, double, char, User, String, Employee)
4. No Arg no return (void)

There is a spl method in class called Constructor.

Constructor is a method, In java, the method name is same as the class name.

Types of Constructor

1. Default or No Arg Constructor
2. Parameterized constructor (Fully/Partially)

A class can have only one default & fully parameterized constructor.

But can have n no of partially parameterized constructor.

3 types of comment available in JAVA

1. Single Line comment // - comments a single line only
2. Multi-line comment /\* continue for many line \*/
3. Documentation comment /\*\* \*/ -- API documentation

Java won’t support explicit pointer manipulation.

OOP - Object Oriented Programming (Everything as a Object) – state & behaviors

POP – Procedural Oriented Programming

1. Abstraction – Hiding the internal Things (Implementation) – Abstract Class & Interface
2. Encapsulation – Hiding the Data – with the help of access modifier.
3. Inheritance – It helps to create relationship between classes. (Father -child,) using extends keyword.
4. Polymorphism – Many forms with single name. [Static /compile-time & Dynamic/runtime] static = method overloading dynamic= method overriding.

In Interface all members are public and methods are abstract.

Interface can have n no of abstract methods, n no of static & default concrete-methods

Abstract-method == Method without body only signature

Concrete- Method == method with body and signature

Two types of interfaces

1. Marker Interface (Interface with no member) – Serializable
2. Functional Interface (Interface with only one abstract method) – Runnable

Access Modifiers

1. Private
2. Default/package
3. Protected
4. Public

Non-Access Modifier (Behaviors)

1. Final – before variable, method, class
2. Static –before variables & methods (Class level member)
3. Abstract – only before class & methods
4. Transient – only for variables
5. Volatile - only for variables
6. Synchronized – for methods and block

Lambda – It’s anonymous function/method. Is a way to provide functionality for a Functional interface. Lambdas are arrow methods in java.

* Thin arrow syntax (->)

Public void display() {

}

Public void () {

}