Phase 1 Mentor Session

Project :

Core Java

OOPS concept

Exception Handling

Overview of Multithreading

Collection Framework

File handling program

Data structure

Object : object is a any real world entity.

Properties or state -- have

Person

Behavior -- do/does

Bank

Animal

Car

Class : blue print of object or template of object.

class Car {

int wheel;

String color;

float price;

void start() {

}

void appliedGear() {

}

void moving() {

}

void stop() {

}

}

Encapsulation : Binding or wrapping data (variable ) and code (function or methods) in a single unit is known as Encapsulation.

class Employee {

private int id;

private String name;

private float salary;

void display(){

}

}

Employee emp = new Emloyee();

emp.salary = -12000;

Polymorphism : One name many forms.

2 types

Compile time polymorphism : static binding or early binding

Method overloading :The method have same name different parameter list. Type of parameter list or number of parameter list is known a method overloading. We can achieve method overloading in same class.

Run time polymorphism : late binding or dynamic binding

Method overriding :The method have same name and same method signature (number of parameter list, type of parameter list and return type must be same). To achieve overriding we need inheritance concept. Super class and sub class required.

Inheritance : Inheritance is use to inherits the properties and behavior of old class to new class.

To achieve inheritance we have to use extends keyword.

class OldClassName {

properties

behavior

}

class NewClassName {

properties

behavior

}

Types of inheritance

1. Single inheritance : one super class an one sub class
2. Multilevel inheritance e : one super class and n number of sub classes connected one by one
3. Hierarchical inheritance : one super class and n number of sub classes connected directly to super class.
4. Multiple inheritance : more than one super class and one sub class. Java doesn’t support this type of inheritance directly we can use this type of inheritance using interface.