JAVA OOPs Concepts (Object-Oriented Programming System)

It is primarily having below crucial points. Without below essential points, we will never be able to achieve OOPs in java, PHP, C#, etc. Now let us see the high-level understanding of OOPs concepts.

- Object --> is nothing, but it is a real-world entity which we can see or feel. In other words, we can say object have shape, behaviour and from which we can perform some task. For ex: Car, Pen, book, etc. (Object create at runtime and reside in heap memory)
- Class --> in simple words we can say it is a blueprint from which objects can be created. (It contains variables and methods in it) For ex:

```
public class TestCLass1{
      //This is a java class
    //It can have variables and techniques and some logics
 }
```

- Inheritance--> it is parent-child relationship. A child will acquire all properties from its parent. In other words, child class will acquire all behaviour and properties from the parent class. Importance of keywords in inheritance is (extends, implements). There are 3-4 types of inheritance exist. (IS-a and HAS-a relationship). Do connect with me will explain you in detail followed by real coding on IDEs.
- Polymorphism--> in simple words we can say it is “performing the same task in different ways”. Poly means multiple and morphism means forms in whole (various forms). Two most import topic under this method overloading and method overriding.
- Abstraction-->is data hiding. It is hiding of real implementation and only showing the functionality. For ex: downloading .exe file, it shows only installation not real code behind that .exe file.
- Encapsulation-->Wrapping of data in a single unit. Java class is one of the encapsulation. For ex: just like different medicine wrapped under on capsule is also encapsulation.

```
        public class TestClass { //Here private sTest is wrapped under
TestClass

              private String sTest="Hi";

              public String getsTest() {
                return sTest;}
            public void setsTest(String sTest) {
                this.sTest = sTest;
        }}
```

Please add on comments and like this lesson, if it gave you some revision or refreshment on OOPs concepts.