1/4/20 **OOP’s**

Object Oriented Programming is a special feature of Java. It is a methodology to design programs using classes and objects. An **object** is anything that has an identity, a behaviour and a state. A **class** is a collection of objects that have the same attributes. There are four different concepts to implement this type of programming. There are:

1. **Inheritance**: It is the property of an object to inherit/acquire various attributes from its parent object.

**Example** – If we take a product, it has the attributes – name, product ID, and price.

Now mobile is a product with the attributes of a product as well as some other attributes like operating system and ram size. So, a mobile inherits the attributes of a product as well as some other distinct attributes.

**Keyword** – extends

**class** Subclass **extends** Superclass{

//methods and fields

}

**Different types of Inheritance:**

* Single Inheritance: Inheritance from one Parent class to one child class
* Multilevel Inheritance: Inheritance from a Parent class to child class to another child class
* Hierarchical Inheritance: Inheritance from one parent class to many other child classes.

1. **Polymorphism**: It is a property of an object which allows it to take multiple forms.

Example – A table can have a different forms/uses. It can be used as a dining table, study table, coffee table, kitchen table, bedside table and so on..

**Different types of polymorphisms:**

* Compile-time: Overloading of a function/method (same method name but different parameters).
* Run-time: Overriding of a function/method (same method name and same parameters and input but access level is less restrictive then the overridden method).

1. **Abstraction**: It is the methodology of hiding the non-essential parts of the data and showing the essential part.

**Example** – If a person is riding a bike he just needs to know how to use the accelerator, brakes, lights but does not need to know how the engine works or how a particular gear falls into place.

**Different types of abstraction:**

* Abstract class: Using keyword ‘abstract’.
* Interface: A blueprint of classes contain various abstract classes.

1. **Encapsulation**: It is the process of wrapping up of data into a single unit. It is achieved by declaring variables as private and using setters() and getters() to access the variable values.

**Collection**

A collection represents a single set of objects, or a group of objects. It is a framework that provides an architecture to store, use and manipulate groups of an object. For Example, it has a framework to sort a list of names, or search a name from a list, or even insert a value in a list.