**Inheritance**

Inheritance is use to inherits or acquire the properties and behaviour of old class to new class.

class OldClass { super class or base class or parent class

property

behaviour

}

class NewClass extends OldClass{ sub class or child class or derived class.

property

behaviour

}

Types of inheritance

1. Single inheritance : one super class and one sub class

Class A { }

Class B extends A {}

1. Multilevel inheritance: one super class and n number of sub classes connected one by one

Class A { }

Class B extends A { }

Class C extends B {}

Class D extends C {}

1. Hierarchical inheritance : one super class and n number of sub classes connected directly to super class.

Class A { }

Class B extends A { }

Class C extends A {}

Class D extends A {}

1. Multiple inheritance : more than one super class and one sub class.

Class A {}

Class B {}

Class C extends A,B{} Java doesn’t support this type of inheritance. Means we can’t extends more than one class at the same time. This type of inheritance we can achieve using interface.

**Oops relationship**

**Is a relationship**

**Has a relationship**

Employee has a Address

Manager/Developer is a type of Employee

Class Employee {

Id,name,salary

Address add = new Address();

}

Class Manager extends Employee{

numberOfEmp;

}

Class Developer extends Employee{

Projectname;

}

Class ProjectManager extends Manager{

clientName

}

Class Address {

City and state

}

**Scanner** is a pre defined class part of util package. which help to take the value through keyboards.

**Polymorphism :** One name many forms or many implementations.

2 types

1. Compile time polymorphism or static binding or early binding

Method overloading: method have same name but different parameter list ie type of parameter list or number of parameter list must be different.

1. Run time polymorphism or dynamic binding or late binding

Method overriding : method have same name and same signature means number of parameter list, type of parameter list as well as return type must be same.

**Annotation**

Meta data: data about data. Java provided lot of pre defined annotation all annotation start with **@** followed by name of the annotation. Few annotation we can use on class level or method level or property level.

**Non access specifiers**

abstract

final

static

**Access specifiers**

private

default (nothing)

protected

public

abstract : abstract is a keyword we can use with method and class but not with variable.

1. abstract method : method without body or incomplete method or without curly braces is known as abstract method.
2. if class contains one or many abstract method then we need to declare that class as abstract class.
3. whichever class extends abstract class that class must be provide the body for all abstract method belong to that class.
4. abstract class can contains normal as well as abstract methods ie it can contains **0** or 1 or many.
5. Abstract class we can’t create the object.
6. Abstract class can contains default as well as we can write parameter constructor.