**Day 5:**

**Encapsulation:** Binding or wrapping data(variables) and code(methods) in a single unit is known as encapsulation.

Example : class.

Inside class we keep all variable and methods. Without creating object of that class we can’t access those property and behaviour. Encapsulation mainly use to achieve security.

Package : package is a collection of classes and interfaces. Using the package we can organize our classes and interfaces in proper manner. Package is a like a directory or folder.

All instance variable must be private and provide helper method which help to set the value with condition to make data secure.

Java Bean class: while creating Java Bean class we need to follow few rules.

1. Class must be public
2. Each instance variable must be private.
3. Each variable we need to create setter and getter methods.
4. Setter method is use to set the value and getter method is use to get the value.
5. Setter method start with pre-fix set followed by variable name and with condition if you need.
6. Getter method start with pre-fix get followed by variable name and get the value.

Java bean class also known as pure encapsulation class.

**Inheritance:** it is use to inherits the properties and behaviour of old class to new class. to achieve inheritance we need to use extends keyword.

class OldClass { // super class or base class or parent class.

property

behaviour

}

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

property

behaviour

}

With help of sub class object we can access its own property and behaviour as well as super class property and behaviour. But with help of super class we can access only its own property and behaviour.

Types of inheritance

1. Single inheritance: one super 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. In Java we can extends only one class at time. we can’t extends more than one class at same time. this type of inheritance we can achieve using interface. Because of this reason java is not 100% pure object.

**Oops relationship**

1. Manager/Developer is a Employee
2. has a

Inside one class we are creating another class object is known as has a relationship.

class Employee {

id,name,salary

Address add = new Address();

Scanner

readEmp()

add.readAdd();

disEmp();

}

class Manager extends Employee {

numberOfEmp

readMgr()

disMgr()

}

class Developer extends Employee{

projectName;

readDev()

disDev();

}

class ProjectManager extends Manager{

clientInfo;

readPmgr()

disPmgr();

}

class Address {

city

state

Scanner

readAdd()

disAdd();

}