OOPs - Object Oriented Programming

It is a programming paradigm which resolves around the concept of object. Object can be considered as real world instance of entities like class, that contain some characteristic & behaviour specified in class template.

→ OOPs helps to understand Software easily and it also maintain DRY Principle

- Class - It is a blue print having data members and member function. eg- CAR, FORM
- Object - Instance of a class having state & behaviour. Memory is allocated only after object instantiation.

4 - Pillars of OOPs

1. **Abstraction** :- Hinding internal details

2. **Encapsulation** :- Wrapping data members and member function into single unit.

3. **Inheritance** :- When the object acquire all the properties and behaviour of parent object

4. **Polymorphism** :- When same object acts differently under different Conditions.

Creating OWN JAVA Class -

Syntax - class <Class_name> {

```
    Field;
    Method;
}
```

eg -

```
class Employee {
    int id;
    int salary;
    string name;
    public void printDetails() {
        Sout("My id is" + id);
        Sout("and my name is " + name);
    }
    public int getSalary() {
        return salary;
    }
}
```

```
Employee ajay = new Employee();
ajay.id = 1;
ajay.salary = 3400;
ajay.name = "ajay";
```

Access Modifiers

Access Modifiers specify where a property/method is accessible. There are four types of access modifiers in java :

1. private
2. default
3. protected
4. public

Access Modifier	within class	within package	outside package by subclass only	outside package
public	Y	Y	Y	Y
protected	Y	Y	Y	N
Default	Y	Y	N	N
private	Y	N	N	N

Constructors -

→ used to initialize an object

→ do not have return type.

→ Whenever we create an object a constructor is called.

Syntax - <Class_name>{

```
    // code
}
```

Types :- ① Default Constructor

② Parameterized Constructor.