Java programming language - OOPS

Data Abstraction
Inheritance
Classes
Objects
Interfaces

Platform independant

JRE - Java Runtime Environment
JSDK/JDK - Java Software Development Kit
JVM - Java Virtual Machine

Classes -- Objects

```
public class Employee{
        private String firstName;
        private String lastName;
        private int empId;

        constructors - Initialize the Objects
        public Employee(){
                this.firstName="N/A";
                this.lastName="N/A";
                this.empId=-1;
        }
Employee e1=new Employee();
        public Employee(String firstName,String lastName,int empId){
                this.firstName=firstName;
                this.lastName=lastName;
                this.empId=empId;
        }
Employee e2=new Employee("Ameya","Joshi",007);
        accessors(getters)/mutators(setters)/facilitators(helpers)
        public String getFirstName(){ return firstName;}
        public String getLastName() {return lastName;}
        public int getEmpId(){ return empId;}

        public void setFirstName(String firstName){

                this.firstName=firstName;
```

```
        }
        public void setLastName(String lastName){

                this.lastName=lastName;
        }
        public void setEmpId(int empId){

                this.empId=empId;
        }
        public double calculateSalary(){-------}
        public void printInfo(){}
}
```

Java Enterprise Application --- Multiple Layers

View Layer - UI Layer
Business Layer -- Business Logic
Persistence Layer -- DB functionalities

MVC - Model View Controller -- View Layer/UI layer/Presentation Layer
Services/Facades - Business Layers
DAO - Data Access Objects (Repositories) -- Persistence Layer/DAO Layer

Model Classes
Services
Facades
Controllers
DAOs

Employee
Employee class as Model
EmployeeService class as service
EmployeeFacade class as facade
EmployeeController as controller
EmployeeDAO as DAO
DTO - Data transfer Object

Java Collections Framework
Lists/Sets/Maps/Comparables/Comparators

==========More About JEE Application Development==========

Employee -->Model/POJO->Plain Old Java Object, Java Bean

1. The class must be within a package
2. It must declare the attributes/properties as private
3. It must have atleast default constructor
4. It must provide getters/setters for properties

classes in business Layer/persistence Layer must be interface driven
EmployeeFacade --> interface
EmployeeFacadeImpl --> class implementing EmployeeFacade
EmployeeService --> interface
EmployeeServiceImpl --> class implementing EmployeeService
EmployeeDAO --> interface
EmployeeDAOImpl --> class implementing EmployeeDAO