1. Create three classes

**Faculty** (facultyid, salary)

**FullTimeFaculty** (basic, allowance) inherits class Faculty

**PartTimeFaculty** (hour, rate) inherits class Faculty

Create a method for accepting input in FullTimeFaculty and PartTimeFaculty, but salary should not be accepted. Salary is calculated on the basis of (basic+allowance) for FullTimeFaculty and (hour\*rate) for PartTimeFaculty. Also create method in above classes to display faculty data.

Create another class(say XYZ) for main method and store 2 fulltime and 2 parttime faculty information. Also print their details.

1. Create an Abstract class **Processor** with int member variable **data** and method **showData** to display data value. Create abstract method **process**() to define processing of member **data**.
   1. Create a class **Factorial** using abstract class **Processor** to calculate and print factorial of a number by overriding the process method.
   2. Create a class **Circle** using abstract class **Processor** to calculate and print area of a circle by overriding the process method

Ask user to enter choice (factorial or circle area. Use Processor class reference to achieve this mechanism.

1. Create Interface **Taxable** with members salesTax=7% and incomeTax=10.5%. create abstract method **calcTax**().
   1. Create class Employee(empId,name,salary) and implement Taxable to calculate incomeTax on yearly salary.
   2. Create class Product(pid,price,quantity) and implement Taxable to calculate salesTax on unit price of product.
   3. Create class for main method(Say XYZ), accept employee information and a product information from user and print income tax and sales tax respectively.
2. Explain the importance of toString() and equals() method of the Object class and override them on class Employee(empId,name,salary).
   1. Create class for main method(say XYZ),and accept five employees information and store in an array. Also ensure if entered empId already exist or not (use **equals** method).
   2. Display all employee info using **toString** method;