Assignment-6
Object Oriented Programming Lab
(Java: Inheritance)

1. You are required to write a base class **Student** having details (name, registration number, father's name, address (current address, permanent address), and contact details (phone number, email Id). **UGStudent**, **PGStudent**, and **PhDStudent** extend the **Student** class they have details about the past qualifying degree and marks. Make each attribute protected and provide suitable getter and setter to access attributes.

2. Write another base class **Course** which stores details about the courses registered by a **Student**. **Course** has attributes such as code, name, and credit. **UGCourse** and **PGCourse** extends the **Course** class. The course code follows the following syntax PG-xxxx and UG-xxxx here x is a number which uniquely identifies a course. Make each attribute protected and provide suitable getter and setter to access attributes.

3. Write another base class **Payment** which stores the payment details of the registered **Student**. **Payment** has a single attribute paymentID which is unique for each payment. **PaymentDD**, **PaymentUPI**, and **PaymentNB** extends the **Payment** class, these classes stores details of payment using Demand Draft, UPI, and Net Banking. Add relevant attributes to these classes to store payment details. Make each attribute protected and provide suitable getter and setter to access attributes.

4. Add one instance of each **Course** class and **Payment** class to the student class so that course and payment details of a student can be associated with each student. Add a member function in Student class to print course detail and payment details using student's registration number. Demonstrate the program by creating 10 students and printing their details.

5. Write a class **Employee** having attributes: employeeID, name, department, dob, designation, yearOfJoining, and phoneNumber. Add a function to print details of employee using employeeID. Two classes, i.e., **Faculty** and **OfficeStaff**, extends the **Employee** class. The **Faculty** has a list of subjects and labs taught by them. **OfficeStaff** has a list of skills which stores skill such as Typing, Technician, etc. Each of the above classes has appropriate getter setter for inserting skill in the list and printing the list of skill. Demonstrate above class using a Test class which creates 5 employees of different types and print their details.

6. Classes **HOD**, **DUGC**, **DMPC**, and **DDPC** extends the **Faculty** class (given in the above question). These classes stores detail specific to the Head of the Department (HOD), Department Undergraduate Committee (DUGC) Convenor, Department Master Program Committee (DUGC)Convenor, and Department Doctoral Program Committee (DUGC) Convenor. **SkilledStaff** and **UnSkilledStaff** extends class **OfficeStaff** (given in the above question). Each class has a public method toString( ) which returns their details as a String. Demonstrate above class using a Test class which creates 6 employees of different types and print their details.

7. **Employee** and **ProductionWorker** Classes
Design a class named Employee . The class should keep the following information in fields:

   - Employee name

- Employee number in the format XXX–L, where each X is a digit within the range 0–9 and the L is a letter within the range A–M.
- Hire date

Write one or more constructors and the appropriate accessor and mutator methods for the class. Next, write a class named **ProductionWorker** that extends the **Employee** class. The **ProductionWorker** class should have fields to hold the following information:

- Shift (an integer)
- Hourly pay rate (a double)

The workday is divided into two shifts: day and night. The shift field will be an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the classes by writing a program that uses a **ProductionWorker** object.

8. **ShiftSupervisor** Class
   In a particular factory, a shift supervisor is a salaried employee who supervises a shift. In addition to a salary, the shift supervisor earns a yearly bonus when his or her shift meets production goals. Design a **ShiftSupervisor** class that extends the **Employee** class you created in question 7. The **ShiftSupervisor** class should have a field that holds the annual salary and a field that holds the annual production bonus that a shift supervisor has earned. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the class by writing a program that uses a **ShiftSupervisor** object.

9. **TeamLeader** Class
   In a particular factory, a team leader is an hourly paid production worker that leads a small team. In addition to hourly pay, team leaders earn a fixed monthly bonus. Team leaders are required to attend a minimum number of hours of training per year. Design a **TeamLeader** class that extends the **ProductionWorker** class you designed in question 7. The **TeamLeader** class should have fields for the monthly bonus amount, the required number of training hours, and the number of training hours that the team leader has attended. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the class by writing a program that uses a **TeamLeader** object.