OOPS Questions

1. A trainer trains many trainees on a given technology in this course, which contains many modules – each module is comprised of different units and each unit has many topics.
   a. Identify the different classes from the above problem statement
   b. Identify the relationship between these classes

2. A company sells different items to customers who have placed orders. An order can be placed for several items. However, a company gives special discounts to its registered customers.
   a. Identify the different classes from the above problem statement
   b. Identify the different connections (relationships) between the above classes

3. A customer enters a mobile shop and asked the dealer for a mobile's availability with certain specification, which includes the camera availability with specified pixels, Bluetooth transfer availability, and memory card extendibility. Define a mobile class with the possible specifications and include other specifications too.

4. The COO of a company wants the list of Employees working in his organization. He wants to know the section, designation, date of joining, qualification and skill set of each employee along with his/her employee id. Design the class Employee for the same.

Objects and Classes

1.
   a. Create a class named 'Employee' with the following members
      - Empno
      - Name
      - Department
      - Salary

   In main create a reference variable of type Employee and allocate memory to it. Initialize the data members and also display the data member values.
   b. Make all the members as private in the Employee class and then try to initialize the data members and print the values.
   c. Make member methods to do the task of initializing and displaying the values.
   d. Write proper setter and getter for each data member
   e. Create 1-D array of employee objects of size 10, initialize the data members for each object and then print the same data on the console. Also create a function to determine which employee salary is the highest.

2. Create a class named BankAccount with the following data members
   - accountNumber
   - accountType
   - userName
   - balance

   a. Add a default constructor to the class which initializes the default values to the data members. The values which should be initialized are:
      - accountNumber = 0

- accountType = NA
- username = NA
- balance = 0.0

b. Create another constructor with 2 arguments(accountNumber and accountType) and both member gets assigned from value from the parameter. For the remaining data members, assign default values as given above.
c. Create another 4 argumented constructor and all members get values assigned from the parameters. The instance variables (data members) and the parameter names must be same.
d. Invoke Constructor created in 'step c' from the constructors created in 'step a' and 'step b'.

3. Create a class to hold information about books. Write a function to display information about the book.

   Data members : Title , Author , cost , numberOfBooks

   Create 3 instances of the above class and initialize the members of the class with the data accepted from the user.

   Accept a title and the numberOfBooks required from the user. Find whether the book exists. If it exists, check if the numberOfBooks are sufficient and if so, indicate the total cost the books

4. Write a program which keeps track of the number of objects that are created and display the count in a function called display().

5. Extend the problem statement 1 to generate empno automatically starting from 101.

6. Develop a class named "TaxCalculator" with following data members
   - Double basicSalary
   - Double tax
   - int netSalary

   a. Create a method calculateTax() that should calculate and print the tax. The tax should be calculated as follows

      Tax = 30*basicSalary/100.

   b. Create another method deductTax() that reduces the tax calculated from the basic salary and store it in another instance variable 'netSalary'

7. Assume that a bank maintains 2 kinds of accounts for its customers, one called savings account & the other current account. The savings account provides compound interest & withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should maintain a minimum balance & if the balance falls below this level, a service charge is imposed.
   Create a class **Account** that store customer name, account number, and type of account. From this derive the classes **Curr-acc** & **Sav-acc** to make them more specific to their requirements. Include necessary methods in order to achieve the following tasks.

a) Accept deposit from a customer & update the balance
b) Display the balance
c) Compute & deposit interest
d) Permit withdrawal & update the balance
e) Check for the minimum balance, impose penalty if necessary & update the balance

Do not use constructors. Use methods to initialize the class members.

8. Create a superclass, Student, and two subclasses, Undergrad and Grad.
   The superclass Student should have the following data members: name, ID, grade, age, and address.

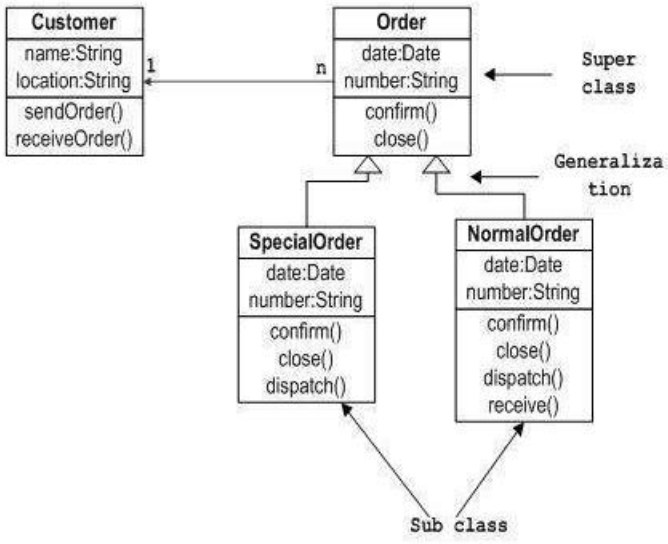
   The superclass, Student should have at least one method: boolean isPassed (double grade)

   The purpose of the isPassed method is to take one parameter, grade (value between 0 and 100) and check whether the grade has passed the requirement for passing a course. In the Student class this method should be empty as an abstract method.

   The two subclasses, Grad and Undergrad, will inherit all data members of the Student class and override the method isPassed. For the UnderGrad class, if the grade is above 70.0, then isPassed returns true, otherwise it returns false. For the Grad class, if the grade is above 80.0, then isPassed returns true, otherwise returns false.

   Create a test class for your three classes. In the test class, create one Grad object and one Undergrad object. For each object, provide a grade and display the results of the isPassed method.

9. Consider the Class diagram given below and create an order application for the same.



   The online order application should be able to receive orders from Customers and process them. There should not be any limit on number/type of orders being placed by the customer.

   Also try considering whether Order can be an abstract class.