

## Core Java Practical Assignments

**NOTE: PLEASE FOLLOW THE NAMING CONVENTIONS STRICTLY AS DISCUSSED DURING SESSIONS.**

### **A. Simulate the stack of integers**

Stack
- data:int[ ] - top:int
+ Stack():constructor + push(int):void + pop():int + peek():int + isFull():boolean + isEmpty():boolean + printStack():void

## B. Simulate the Linear Queue for integers

Queue
- data:int[ ] - rear:int - front:int
+ Queue():constructor - isFull():boolean - isEmpty():boolean + insert(int):void + delete():int + peekQ():int + printQ():void

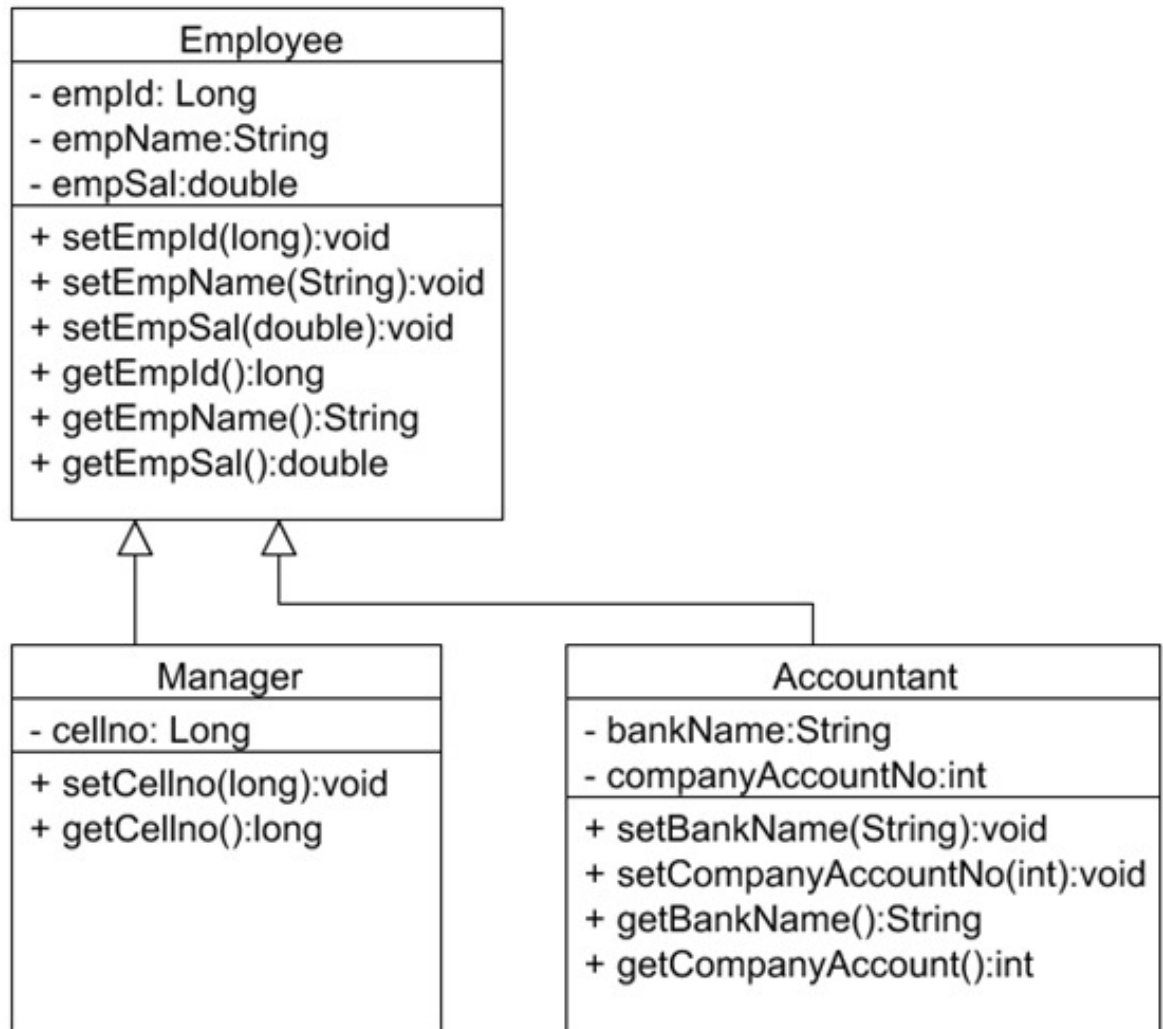
Use the above class to simulate the circular Queue of integers

**C. Create a class Account as shown below.**

**Write a class AccountTest to demonstrate the use of account class.**

Account
- acType:String - balance:double
+ Account() + Account(acType:String,balance:double) + setAcType(acType:String):void + getActype():String + getBalance():double + withdraw(amount:double):void + deposit(amount:double):void + printAccountInfo():void

**D. Develop the program for Employee Management system with following class Hierarchy.**



**Add the following subclasses to the existing Super class Employee**

**1. AreaManager**

**int areaCode;**

**String areaName**

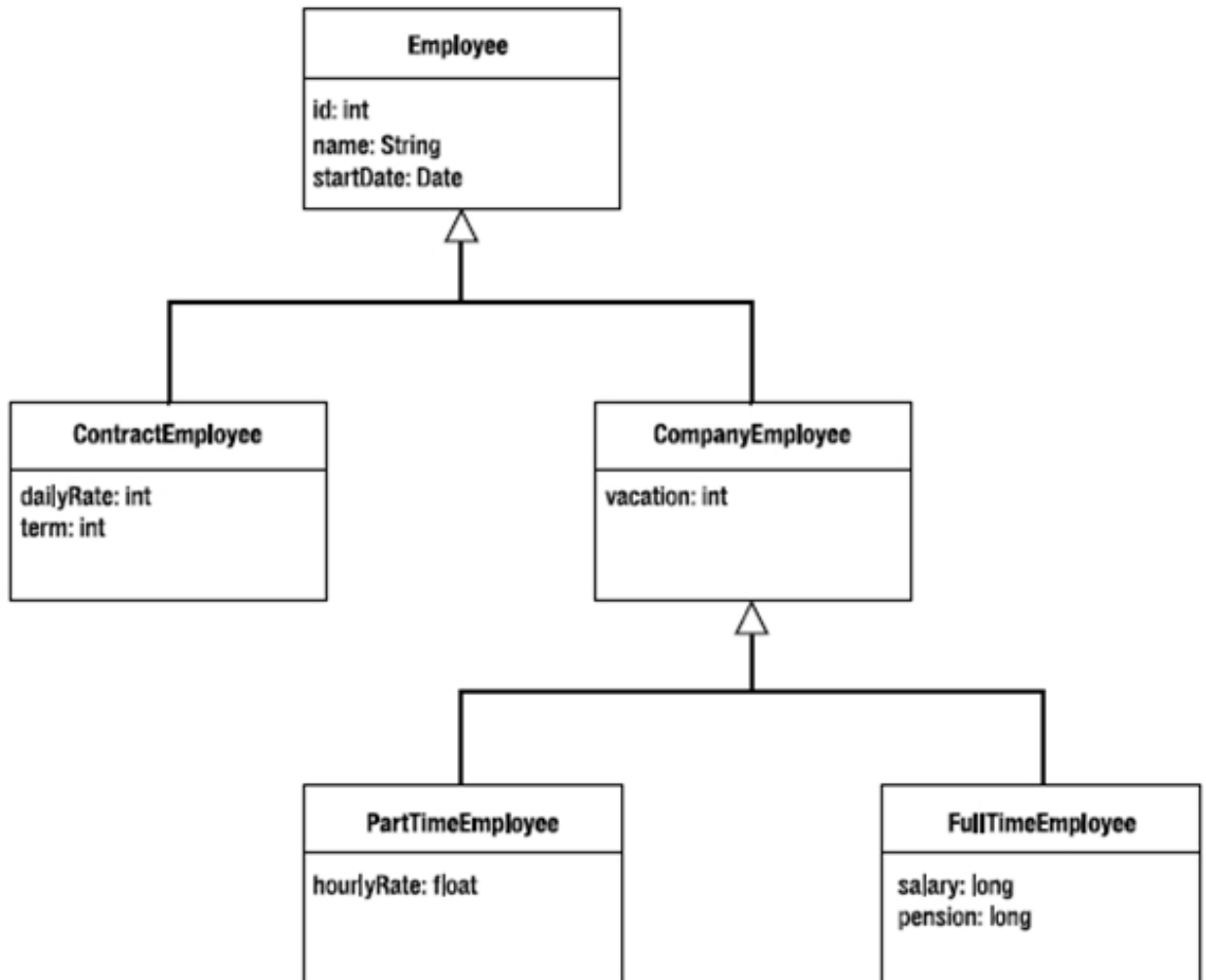
**2. SalesPerson**

**double travelingAllowance**

**double dearnessAllowance**

## E. Employee Management System

Develop a program for Employee Management System using the following Hierarchy



**F. Construct the Appropriate UML class diagrams for the below mentioned classes. Also develop a program to implement the classes.**

**class Vehicle**

**attributes**

**registration no  
weight  
price**

**methods**

**constructors  
setXXX/getXXX**

**class TwoWheeler inherited from Vehicle**

**attributes**

**color**

**methods**

**constructors  
setXXX/getXXX**

**class FourWheeler inherited from Vehicle**

**attributes**

**model**

**methods**

**constructors  
setXXX/getXXX**

**class Low inherited from FourWheeler**

**attributes**

**capacity**

**methods**

**constructors  
setXXX/getXXX**

**class medium inherited from FourWheeler**

**attributes**

**capacity**

**methods**

**constructors**

**setXXX/getXXX**

**class heavy inherited from FourWheeler**

**attributes**

**capacity**

**methods**

**constructors**

**setXXX/getXXX**

**G. A bank accepts fixed deposit for one year or more and the policy it adopts on interest is as follows:**

**i) If a deposit is less than Rs.2000 and for 2 or more years,**

**the interest rate is 5% compound annually.**

**ii) If a deposit Rs.2000 or more but less than Rs.6000 and for 2**

**or more years, the interest rate is 7% compound annually.**

**iii) If a deposit is more than Rs.6000 and for 1 or more years,**

**the interest rate is 8% compound annually.**

**iv) On all deposits for 5 year or more, interest is 10% compound**

**annually.**

**v) On all other deposits not covered by above conditions, the**

**interest is 3% compound annually.**

**Note: Compound Interest =  $p \cdot (1 + r/100)^T$  to the power T.**

**Give the amount deposited and the number of years, write a program to**

calculate the money in the customer's account at the end of the specified time.  
(HINT:Use static methods)

**H. Draw the class diagram for the below mentioned class**

**Write a class to validate date.**

**Attributes**

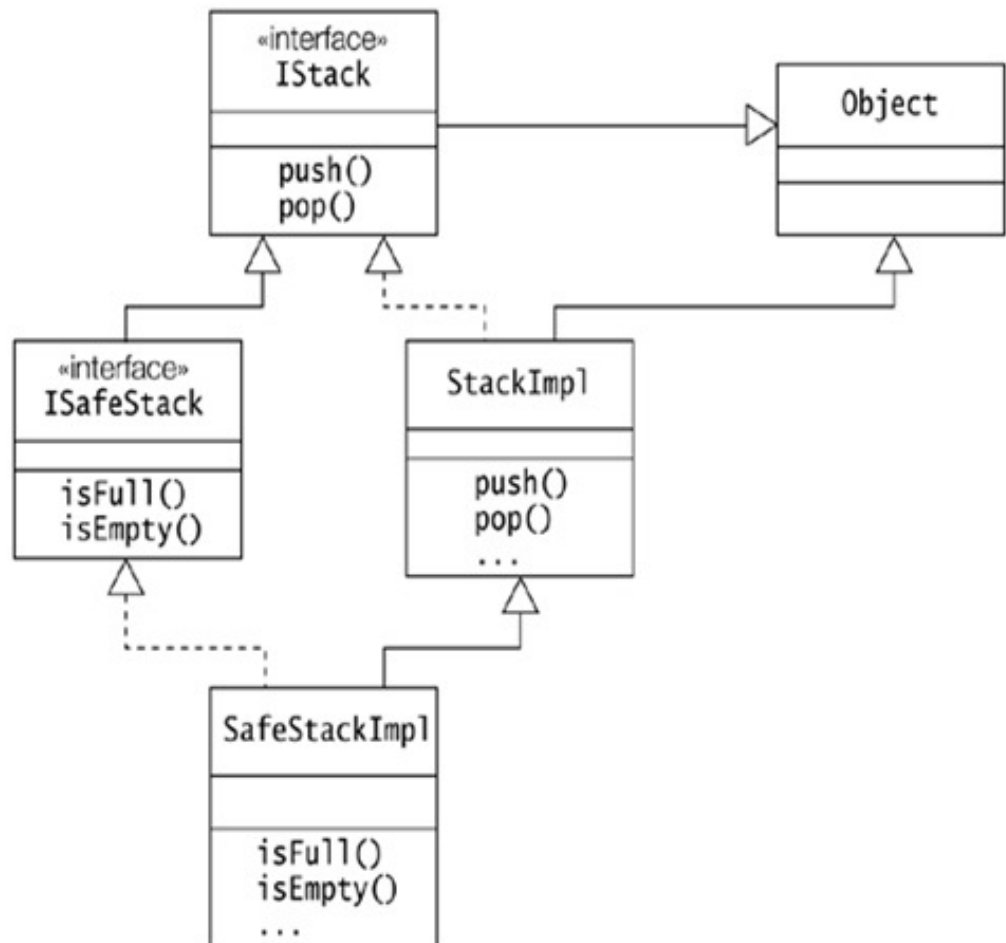
**day  
month  
year**

**Methods**

**incrementByOneDay( ) //increment the date by one day  
decrementByOneDay( ) //decrement the date by one day  
compare( ) //will compare 2 dates & return true/false accordingly  
Initialize the data members through constructors only.**



**I. Study the UML Diagrams below and develop a program to create the Improved Stack**

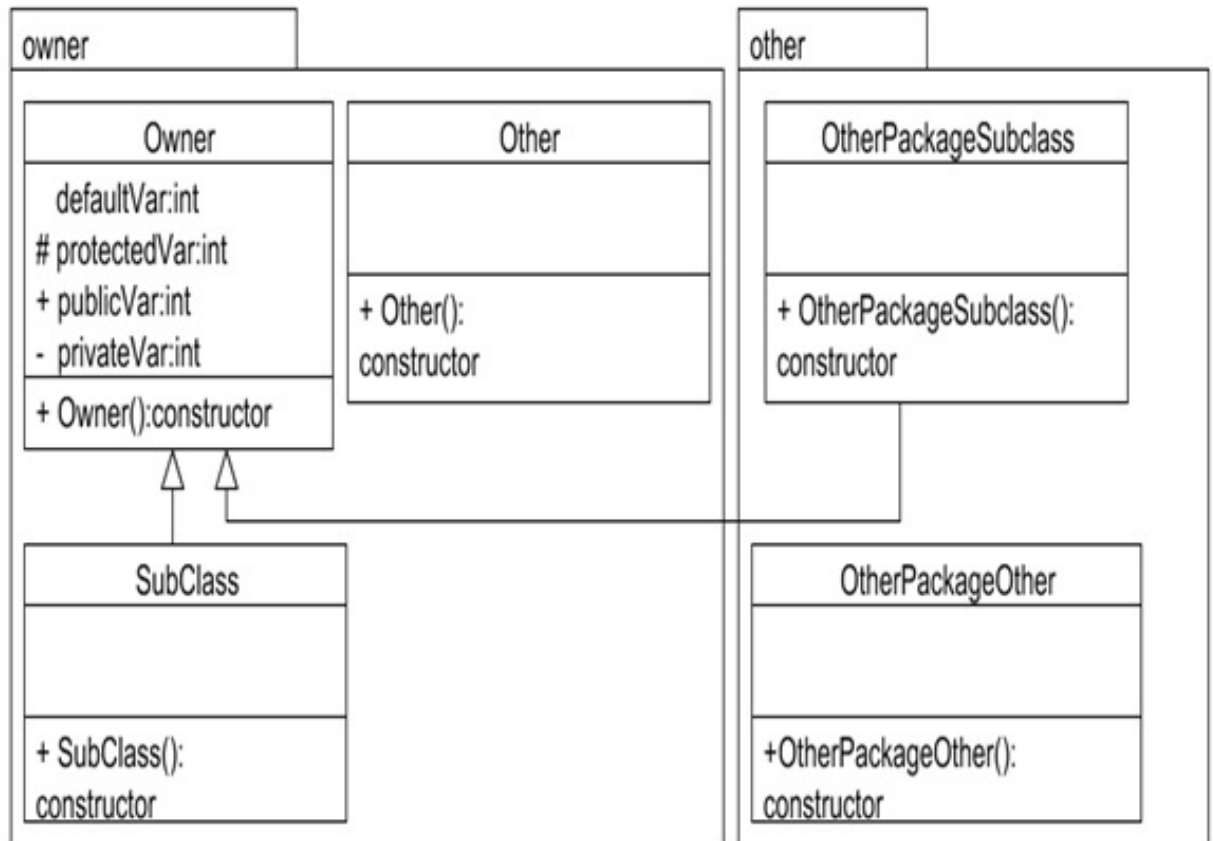


**J. Explore the Marker interface Cloneable, and develop a program to clone an object of class CloneableClass.**

**K. Implement above UML structure for implementing Linear/Circular Queues(Draw appropriate Class Diagrams)**

**L. Override the toString()/equals() methods for the Employee/Date/Stack/Queue classes.**

**M. Implement and test the below given package class Hierarchy**



**N. Re design the assignment F such that all the classes fall in their respective packages. Follow the naming conventions strictly as discussed in the sessions.**

**O. Draw the UML diagram for the following scenario and develop the application as stated**

**Create a package, customer to contain an abstract base class customer**

**attributes**

**name**

**address**

**methods**

**set method to set the values of the data members**

**create an interface in the same package i.e. customer**

**interface account**

**methods**

**deposit**

**withdraw**

**checkbalance**

**A class SavingsAccount, in a separate file & separate package say savings is saved outside customer. This class should inherit from the above interface & the abstract class**

**attributes**

**balance**

**amount**

**account\_no**

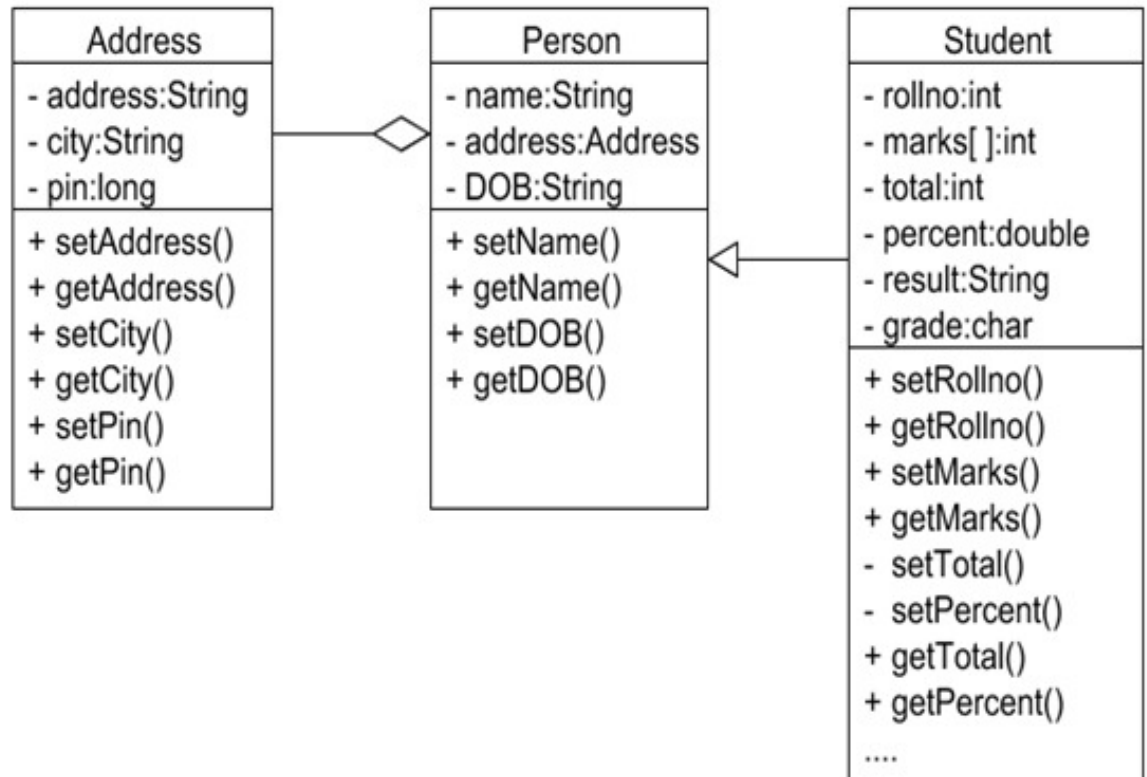
**date\_of\_acct\_opening**

**methods**

**set method to set the values of the data members**

**methods to be overridden.**

**P. Implement following classes and design student marksheet generation system. Implement all the classes in different packages.**



**Q. Implement the classes as described below**  
**Use the appropriate Collection Object to store the Shape Objects.**

