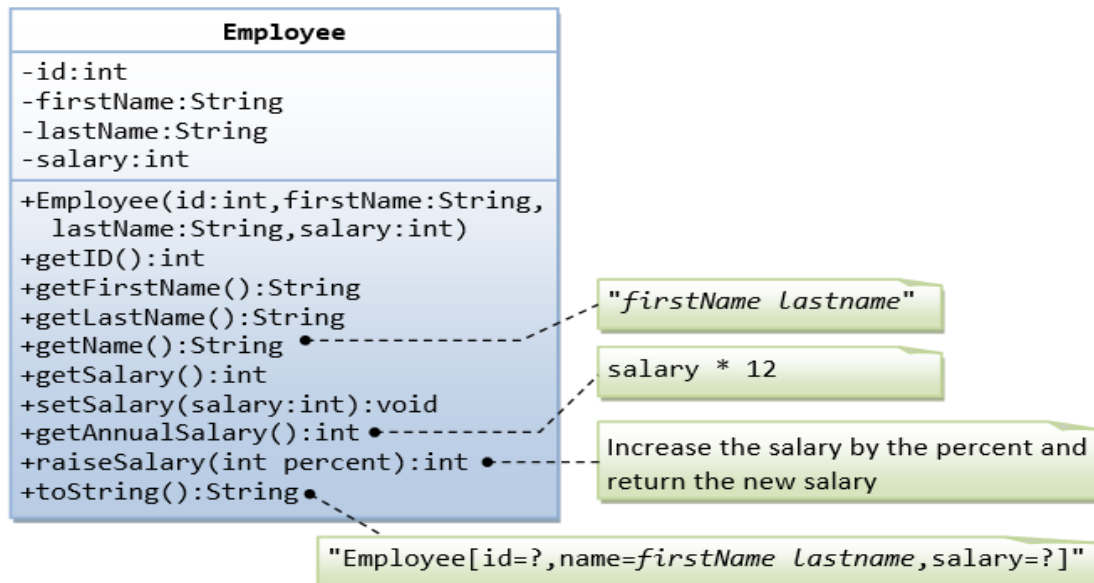


1. A class called Employee, which models an employee with an ID, name and salary, is designed as shown in the following class diagram. The method raiseSalary(percent) increases the salary by the given percentage. Write the Employee class.



Below is a test driver to test the Employee class:

```

public class TestMain {
    public static void main(String[] args) {
        // Test constructor and toString()
        Employee e1 = new Employee(8, "Amit", "Jain", 2500);
        System.out.println(e1); // toString();

        // Test Setters and Getters
        e1.setSalary(999);
        System.out.println(e1); // toString();
        System.out.println("id is: " + e1.getID());
        System.out.println("firstname is: " + e1.getFirstName());
        System.out.println("lastname is: " + e1.getLastName());
        System.out.println("salary is: " + e1.getSalary());

        System.out.println("name is: " + e1.getName());
        System.out.println("annual salary is: " + e1.getAnnualSalary()); // Test
method

        // Test raiseSalary()
        System.out.println(e1.raiseSalary(10));
        System.out.println(e1);
    }
}

```

The expected output is:

```

Employee[id=8,name=Amit Jain,salary=2500]
Employee[id=8,name=Amit Jain,salary=999]
id is: 8
firstname is: Amit

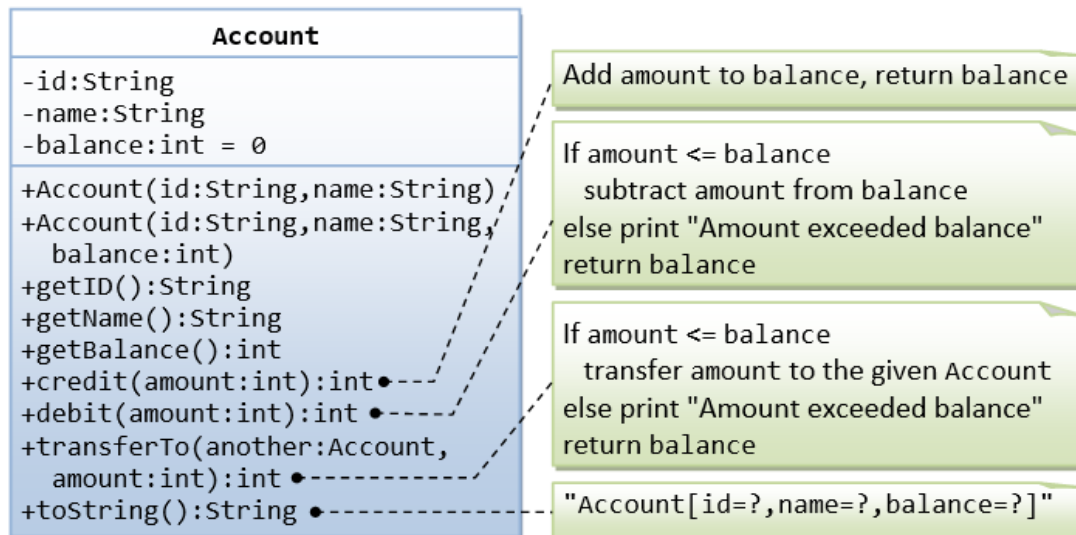
```

```

lastname is: Jain
salary is: 999
name is: Amit Jain
annual salary is: 11988
1098
Employee[id=8,name=Peter Tan,salary=1098]

```

2.



Below is a test driver to test the Account class:

```

public class TestMain {
    public static void main(String[] args) {
        // Test constructor and toString()
        Account a1 = new Account("A101", "Sharma", 88);
        System.out.println(a1); // toString();
        Account a2 = new Account("A102", "Kumar"); // default balance
        System.out.println(a2);

        // Test Getters
        System.out.println("ID: " + a1.getID());
        System.out.println("Name: " + a1.getName());
        System.out.println("Balance: " + a1.getBalance());

        // Test credit() and debit()
        a1.credit(100);
        System.out.println(a1);
        a1.debit(50);
        System.out.println(a1);
        a1.debit(500); // debit() error
        System.out.println(a1);

        // Test transfer()
        a1.transferTo(a2, 100); // toString()
    }
}

```

```

        System.out.println(a1);
        System.out.println(a2);
    }
}

```

The expected output is:

```

Account[id=A101,name=Sharma,balance=88]
Account[id=A102,name=Kumar,balance=0]
ID: A101
Name: Sharma
Balance: 88
Account[id=A101,name=Sharma,balance=188]
Account[id=A101,name=Sharma,balance=138]
Amount exceeded balance
Account[id=A101,name=Sharma,balance=138]
Account[id=A101,name=Sharma,balance=38]
Account[id=A102,name=Kumar,balance=100]

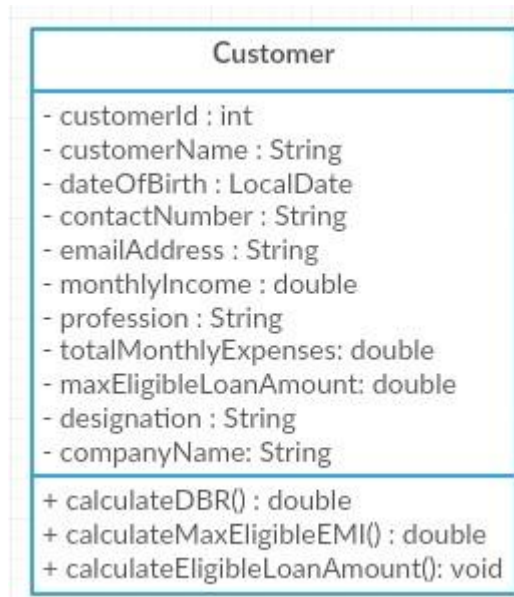
```

3. In the above class Account,
 - a. Add a default constructor to the class which initializes the default values to the data members. The values which should be initialized are:
 - id = "0"
 - name = 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. In the 3 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'.
4. Create a class Student with following data members:
 - String studentId
 - String studentName
 - Int creditPoints
 - a. Create a 2-arg constructor taking studentName and studentId as parameter. The constructor should validate the parameters (name should be minimum 4 characters and id should be minimum 3 characters long). If the validation is not successful, print an error message and assign default values to the data members.
 - b. Create a method getLoginName() which displays the login name for the student (login name will be the combination of first 4 characters of the student's name and first 3 characters of the student's id).
 - c. Create a method changeStudentName(String) for changing the student's name with a new name.

- d. Create `getStudentName()` and `getStudentId()` methods to return student's name and student's id.
- e. Create a method `addCredits(int)` which adds the new credit points to the existing credit points.
- f. Create a `getCredits()` method which displays the added credit points.
- g. If you invoke `addCredits()` and then `getCredits()`, the displayed credit points will be the sum of all the points added till now.

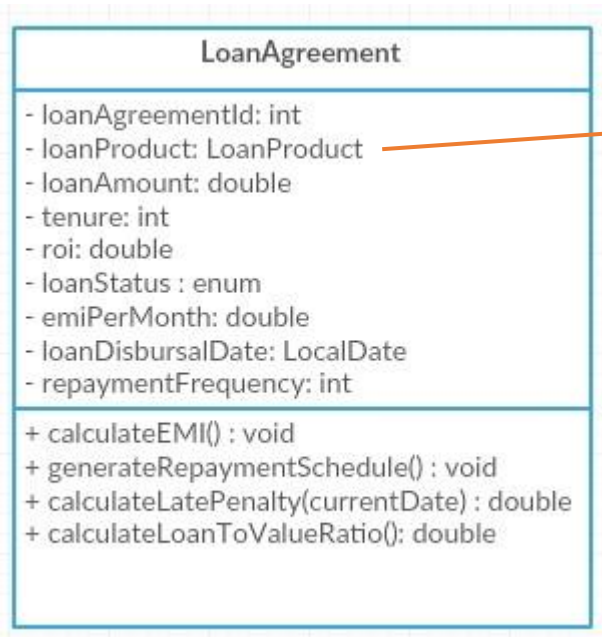
Create 5 instances of Student class and initialize the members of the class with the data accepted from the user. Perform the above mentioned operations on each instance and display the details.

5. Create a class Customer according to the given class diagram.



The methods functionalities are as per the Day – 1 assignment.

6. Create a class Loan according to the given class diagram



Create LoanProduct as an Empty Class as of now.

The method formulas are given in Day-1 assignments.