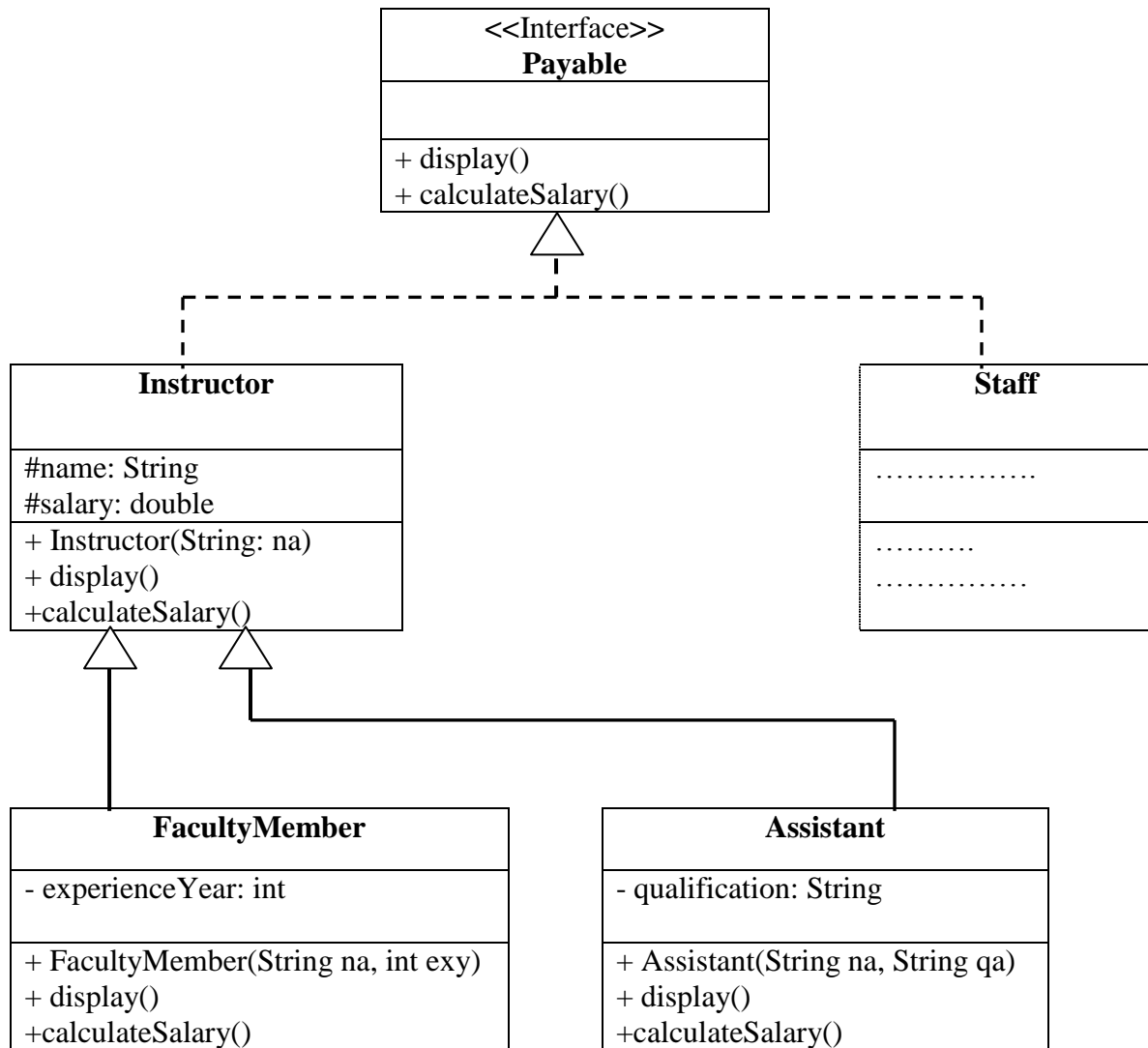


Name:	Section:	Spring 2010
ID:	CSC113 Final Exam	Dr.

**Question1:** Consider the following UML class diagram



The interface **Payable** contains the following methods:

- *display()*
- *calculateSalary()*

The class **Instructor** contains the following attributes and methods:

- name : name of the instructor
- salary: monthly salary of the instructor
- *display()* : displays all the attributes of the Instructor
- *calculateSalary()*

The concrete class **FacultyMember** contains the following attributes and methods:

- experienceYear: number of experience years
- *display()* : displays all the attributes of the faculty member
- *calculateSalary()*

Note that the salary of a faculty member is calculated as follows

$$\text{salary} = 5000 + \text{experienceYear} * 300$$

Write in Java the interface **Payable**, the class **Instructor**, and the concrete class **FacultyMember**.

Note: You can call getters and setters without implementation. Assume that the class Staff and Assistant are implemented.

**Question2:** Consider the following Java program. Trace and write the output of the program.

```
public class Test
{
    public static void main(String args[])
    {

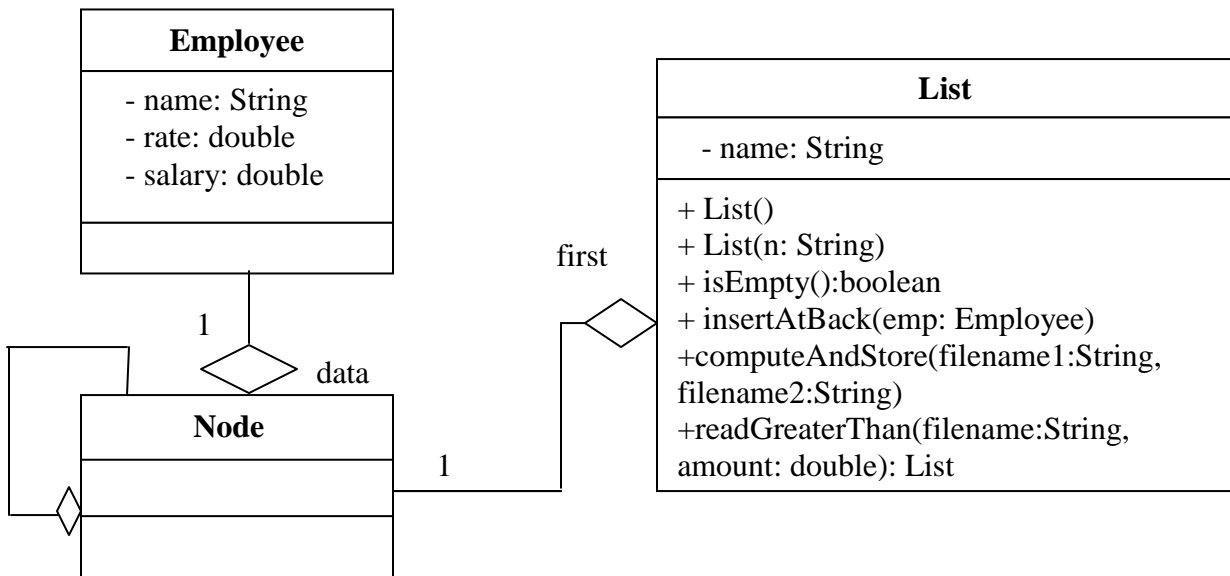
        int numer[] = { 4, 8, 16, 32, 64, 128, 256, 512 };
        int denom[] = { 2, 0, 4, 4, -1, 8 };

        try
        {

            for(int i=0; i<numer.length; i++)
            {
                try
                {
                    System.out.println(numer[i] + " / " + denom[i] + " is " +numer[i]/denom[i]);
                }
                catch (ArithmeticException exc)
                {
                    System.out.println("Can't divide by Zero!");
                }
            }
        }catch (ArrayIndexOutOfBoundsException exc)
        {
            System.out.println("Index out of bounds.");
            System.out.println("Fatal error .");
        }
    }
}
```

Name:	Section:	Spring 2010
ID:	CSC113 Final Exam	Dr.

**Question3 :** We want to manage a company of employees, for that we consider the following UML class diagram.



**Class List:** The class List contains the following attributes and methods:

- name : name of the List
- first : first Node of the List

**public void computeAndStore(String filename1, String filename2)**

This method receives a binary file called *filename1* containing the worked hours done by the employees. The method reads the content of *filename1* and for each employee of the current linked list, it calculates the salary like follows:

salary = worked hours \* rate

(rate is an attribute of the Employee class which is the hourly rate)

The method then stores all the Employee objects in the file *filename2*.

Hint: the length of the linked list and the length of the binary files are equal.

**public List readGreaterThan(String filename, double amount)**

This method reads Employee objects from the object file *filename* and returns a linked list containing only employees having salary greater than amount.

Write in Java the two methods *computeAndStore* and *readGreaterThan*. Consider that the other methods of the class List are implemented.

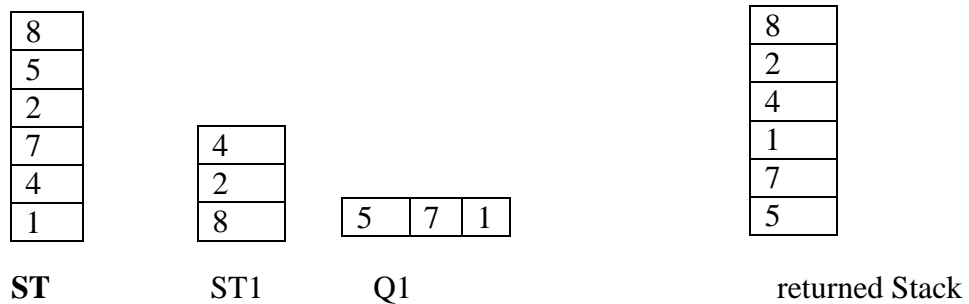
**Answer :**

Name:	Section:	Spring 2010
ID:	CSC113 Final Exam	Dr.

**Question 4 :** We want to reorder elements in a stack. For that we should implement the following method: **public Stack reorder( Stack ST)**

This method receives a stack **ST** of integer numbers and returns a new stack containing odd numbers in the bottom and even numbers at the top. You should use temporary stack and queue as shown in the example to solve the problem.

**Example:**



Hint: Assume that the methods of the class Stack and the class Queue are implemented.

**Answer :**