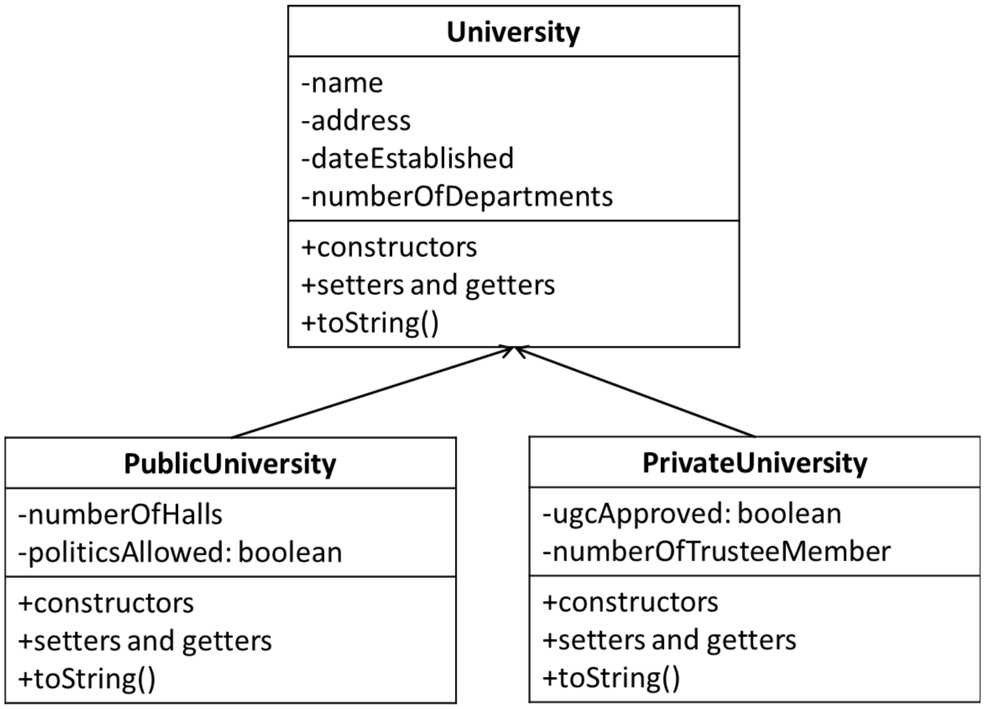
# **United** International **University**

## Department of Computer Science and Engineering

### CSE 1116: Object Oriented Programming Lab, Assignment3

---

1. Consider the following class diagram. University is a superclass, and PublicUniversity and PrivateUniversity are two extended classes derived from the University class.



a. 2. 2.Create the required classes with appropriate constructors, setters, getters, and toString methods. Use appropriate data types from your knowledge.

b. In the main method, create an ArrayList of University objects named *universities*. Create three PublicUniversity objects and three PrivateUniversity objects (any order) by taking inputs from the user and insert them into the *universities*.

c. Sort the *universities* ArrayList based on their established year in descending order. To do this, you might need to modify the University class (hint: comparable interface).

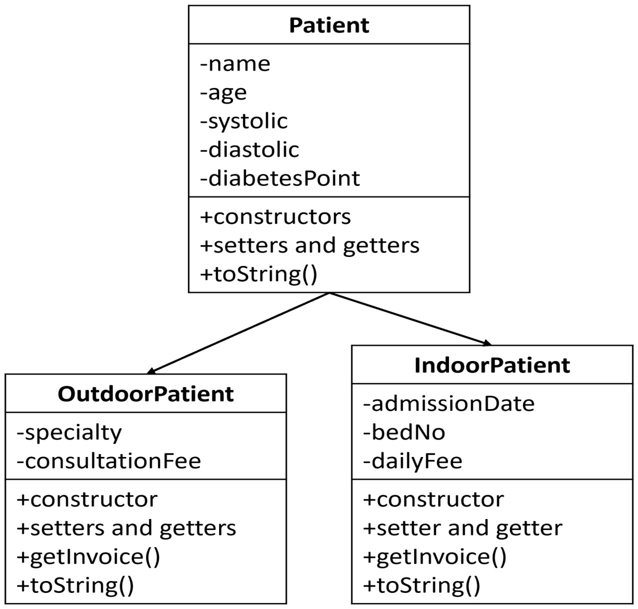
d. Print all the universities that are approved by UGC.

e. Print all the universities where politics are not allowed.

f. Take an address as input from the user and display all the universities from that address.

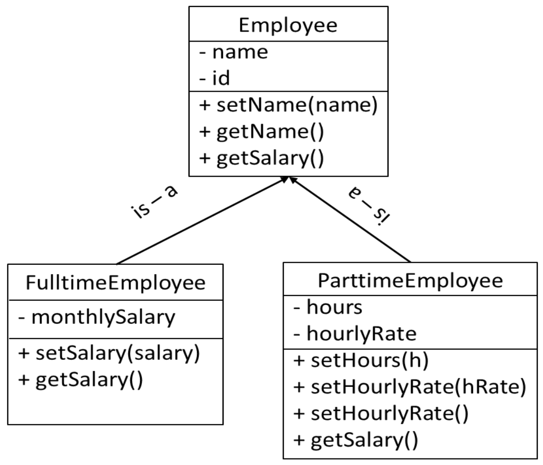
g. Print the details of every university including numberOfHalls, politicsAllowed, ugcApproved, numberOfTrusteeMember.. To see the appropriate information from the universities ArrayList, you need to override a few methods in PublicUniversity and PrivateUniversity classes. If a university is Public, you should print "Public" before printing the details. If a university is Private, you should print "Private" before printing the details.

2. Consider the following class diagram. Patient is a superclass where systolic is the first number (normal is 120) and diastolic is the second number (normal is 80) to measure the blood pressure. OutdoorPatient and IndoorPatient are two extended classes derived from the Patient class.



a. Create the required classes with appropriate constructors, setters, getters, and toString methods. Use appropriate data types from your knowledge.
b. getInvoice() of OutdoorPatient will simply return the consultationFee, however, getInvoice() of IndoorPatient will return after calculating the bill based on how many days the patient was in the hospital and based on the dailyFee.
c. In the main method, create an ArrayList of Patients objects named *patients*. Create three IndoorPatient objects and three OutdoorPatient objects (any order) by taking inputs from the user and insert them into the *patients*. [2]
d. Sort the *patients* ArrayList based on their age in descending order. To do this, you might need to modify the Patient class (hint: comparable interface).
e. Print all the patients who have high or low blood pressure (systolic greater than 120 or diastolic less than 80).
f. Print the details of every patient including their bills. To see the appropriate bill, you need to override the getInvoice() methods. If a patient is an OutdoorPatient, you should print "Outdoor" before printing his/her details. If a patient is IndoorPatient, you should print "Indoor" before printing his/her details.

3. Consider the following class diagram. Employee is a superclass and FulltimeEmployee and ParttimeEmployee are two subclasses. A FulltimeEmployee is paid on a monthly basis whereas a ParttimeEmployee is paid on an hourly basis. The hourly salary of a ParttimeEmployee is negotiable or a minimum wage of 200 TK will be given.



a. Create the required classes with appropriate constructors, setters, and getters. [4]

b. Add an attribute named ***dateOfBirth*** which is a Calendar type instance to the Employee class. Also add corresponding setter and getter and modify the constructor such that it takes day, month and year as parameters and sets the Calendar type instance. [1]

c. getSalary() of FulltimeEmployee will simply return monthlySalary, however, getSalary() of ParttimeEmployee will return after calculating the salary based on how many hours s/he worked and the hourlyRate. [2]

d. In the main method, create an ArrayList of Employee objects named *employees*. Create three FulltimeEmployee objects and three ParttimeEmployee objects (any order) by taking inputs from the user and insert them into the *employees*. Before creating the objects, you need to check whether an employee with the same ID is already inserted into the ArrayList or not. [2]

e. After populating the *employees* ArrayList, print the details of every employee including the salary. To see the appropriate salary, you need to override the getSalary() method of the Employee class. If an employee is a FulltimeEmployee, you should print "Fulltime" before printing his/her details. If an employee is ParttimeEmployee, you should print "Parttime" before printing his/her details. [3]

f. In the main class, write a method that will take a year as parameter and print all the employees' information having that birth year. [1]

g. Sort the ***employees*** ArrayList based on their salary. (hint: use comparable interface). [2]

[Bonus: You can add extra features/menus in your applications]
Submission guideline: Submit a zipped file containing all three applications.