

CS 112 Programming II

Lab Session 5: Inheritance and Polymorphism

Objectives

By now you should have developed your ability of using the Object-Oriented paradigm in approaching programming tasks. By the end of today's lab session you should be able to:

- define a subclass from a superclass through inheritance.
- invoke the superclass's constructors using the `super` keyword.
- override instance methods in the subclass.



Exercise 1

The Person, Student, Employee, Faculty, and Staff Classes

Design a class named `Person` and its *two subclasses* named `Student` and `Employee`. Make `Faculty` and `Staff` subclasses of `Employee`. The details of the classes are described below:

- The `Person` class has the private properties `name` and `email` along with their `accessors` and `mutators`.
 - The `Student` class has the private properties `studentId` and `status` (freshman, sophomore, junior, or senior). It also has a constructor with four arguments: name, email, studentId and status.
 - The `Employee` class has the private properties `employeeId` and `salary` along with their `accessors` and `mutators`.
 - The `Faculty` member has the private properties `officeHours` and `rank` along with their `accessors` and `mutators`. It also has a constructor with four arguments: name, email, employeeId and rank.
 - The `Staff` class has the private property `title` along with its `accessor` and `mutator`. It also has a constructor with four arguments: name, email, employeeId and title.
- Draw the UML diagram for the classes and implement them.
 - Write a test program that creates `Student`, `Faculty`, and `Staff` objects using the constructors you have created and prints their data. A sample run is available in the next page.

Sample Run

```
Student Name: Amal Ahmad
Student Email: Amal@taibahu.edu.sa
Student ID: 3233345
Student Status: Senior
=====
Faculty Name: Sara Abdullah
Faculty Email: Sara@taibahu.edu.sa
Faculty ID: 2341
Faculty Rank: Assistant Professor
=====
Staff Name: Nora Khaled
Staff Email: Nora@taibahu.edu.sa
Staff ID: 1222
Staff Title: Quality Assurance Officer
=====
```



Exercise 2

Instead of using setters within your **Student**, **Faculty**, and **Staff** classes to initialize the data fields of parent classes, use **super** keyword to explicitly invoke parents' constructors. You would need to add constructors within your parent classes (i.e. Employee and Person) to achieve the required task.



Exercise 3

Override the `toString()` method as follows:

- **Student** class: the method returns the student **name** and **studentId**.
- **Faculty** class: the method returns the faculty **name**, **employeeId**, and **rank**.
- **Staff** class: the method returns the **name**, **employeeId**, and **title**.