

## Lab 8: Inheritance, Abstract classes and Interfaces

0. This lab is due by **4pm on Friday, July 1<sup>st</sup>**. You should be checked off by 4pm. Do not wait till 3.45pm on Friday before trying to get checked off.
1. In this lab, you will explore the concepts and procedures involved in using inheritance, Abstract classes and Interfaces.
2. Create an **Interface** called '*Employee*'. Employees typically work 8 hour days but on some days might work more than 8 hours. When they work more than 8 hours, they are paid an overtime wage for the number of hours they work beyond the regular 8 hour work schedule. The employee interface you create should support the following methods (Hint: Think about what arguments each method needs as well as what data type each method should return):
  - **getWage**  
This should simply return the employee's wage per hour.
  - **setWage**  
This should set the employee's hourly wage
  - **getOvertimeWage**  
This should return 1.5 times the regular wage
  - **totalDailyPay**  
This should calculate the total daily wage for the employee (including overtime!).
3. Create an **Abstract Class** called '*Student*'.

This class should have the following concrete and abstract methods:

### Abstract methods:

- **gradeAverage**  
This method should find the average grade of the student. It should take in an array of doubles as input. The student has an 'A' if his average grade is above 85. It should return the character grade of the student. For simplicity, you can make any grade below 85 a B.

### Concrete methods:

- **getName**  
This should return the name of the student

- **getCourse**  
This should simply print the degree the student is pursuing in school (Example: IT, Business Communication, etc...)

4. Create an *AttachmentStudent* class. This class should inherit from the *Student* class and implement the *Employee* interface. The constructor should be as follows:

Constructor

In addition to the requisite inputs for its parents (superclass and interface), it should take in an array that contains 4 grades representing an average of 4 courses the student has taken. What is the type of this array?

In addition to the requisite methods in this class, include the following methods:

- **hired**  
This method returns true if the student has a grade average of an A. Otherwise it returns false
- **setHours**  
This method should first check if the student has been hired or not. It should only proceed if the student has been hired. Otherwise, it should print a message that states – “Student has not been hired. Student does not have the requisite grades.”  
This method should set the number of hours worked. It should include regular hours and overtime hours. What are the appropriate arguments?
- **getHours**  
This should return the number of regular hours worked and the number of overtime hours worked. It should return these two numbers in an array. What is the array type? If the student has not been hired, it should return null.
- **toString**  
This method should print the following:  
Name:  
Course:  
Hired: Yes/No  
Wage:  
Total pay:

5. Create at least 3 Attachment students. Two of these students should be eligible for a Student attachment (i.e. they should have a grade average of an A). Set the

hours worked for the students, their wages, courses etc. Calculate the total pay and call the toString method for all 3 students.

## **6. Checkoff**

Compile and test your program on your own. Call a staff member to your computer and demonstrate a successful run of the code. We may modify your test code and quiz you on various aspects of your lab.

**We will not accept modifications to the required files after the lab is officially due.**

MIT OpenCourseWare  
<http://ocw.mit.edu>

EC.S01 Internet Technology in Local and Global Communities  
Spring 2005-Summer 2005

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.