

CSC 1302 Principles of Computer Science II

Lab 13: Java Inheritance

(Due on 11:59 pm, 7/21/2021)

Purpose:

A class in Java can extend another class to become a subclass. When class B extends class A, B becomes subclass (child) and A becomes superclass (parent). The subclass can reuse all the features of the parent class. An interface defines a set of specifications that other classes must implement. Implementing an interface allows a class to become more formal about the behavior it promises to provide.

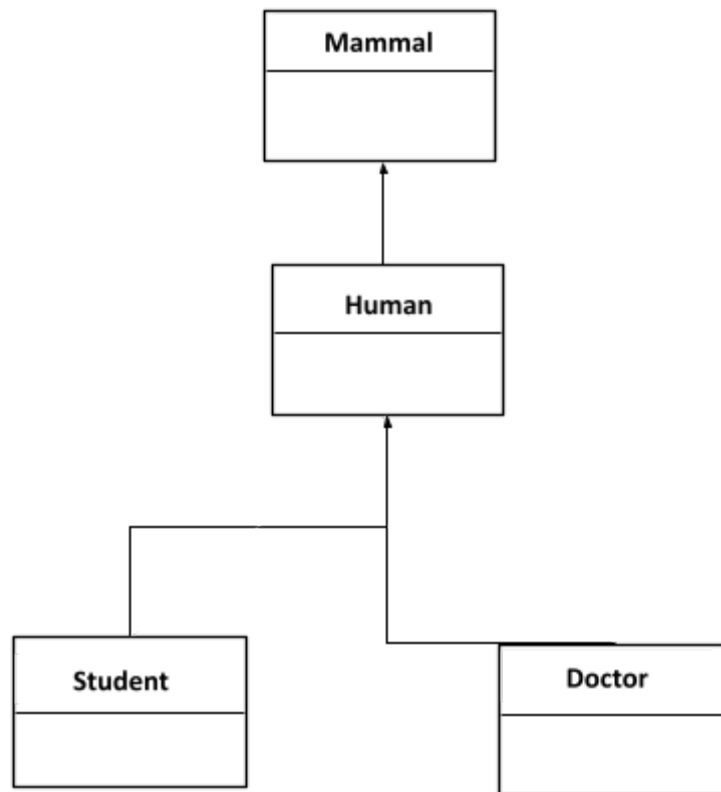
In this assignment, we will practice how to extend classes and create corresponding objects. We will also practice using the interface. The meaning of implementing an interface is not very different than extending a class but it comes with an additional caveat. When a class implements an interface, it has to provide an implementation of all methods declared inside interface.

Criteria:

1. Upload all of the .java and the .class files to the CSc1302 dropbox on <http://icollege.gsu.edu>.
2. Your assignment will be graded based on the following criteria: (a) Are your programs runnable without errors? (b) Do your programs complete the tasks with specified outputs? (c) Do you follow the specified rules to define your methods and programs? (d) Do you provide necessary comments include the programmer information, date, title of the program and brief description of the program.
3. Please comment the important lines in the .java file as shown in the template. The important lines including but not limited to i) variables, ii) for-loop, iii) while-loop, iv) if-else statement, iv) methods. Please use your own words to describe what is your purpose to write this line. A .java file without comment will be graded under a 40% penalty.
4. Make sure that both the .java and .class files are named and uploaded to icollege correctly. If any special package is used in the program, be sure to upload the package too. Should you use any other subdirectory (whatsoever) your program would not be graded, and you will receive a 0 (zero).
5. No copying allowed. If it is found that students copy from each other, all of these programs will get 0.

Task:

The class diagram with four classes **Mammal**, **Human**, **Student** and **Doctor** is given. The **Mammal** class is given as well. Write down the remaining classes as described in the class diagram. The fields and methods for each class is given below.



1. **Class Human:**

- a. Fields: age (int), weight (double), height (double)
- b. Methods: getAge, getWeight, getHeight

2. **Class Student:**

- a. Fields: major (String), gpa (double), creditHours (int)
- b. Methods: getMajor: returns major
getGpa: returns gpa
getYear: returns freshman, sophomore, junior or senior as determined by earned credit hours:
 - Freshman: Less than 32 credit hours
 - Sophomore: At least 32 credit hours but less than 64 credit hours
 - Junior: At least credit hours SH but less than 96 credit hours

- Senior: At least 96 credit hours

3. **Class Doctor:**

- a. Fields: years (int), Speciality (String)
- b. Methods: getYears: returns years
getSpecialty: returns speciality
getSalary: calculates salary. The base salary is 40,000 for the 1st year and for each additional year gets 5000 more. So, in second year the doctor gets 45,000, in third year he gets 50,000 and so on.

```
//In Mammal.java
Public interface Mammal
{
    public double hairColor();
}
```

4. Write another client class.

- a. Create an object of **class Student**
 - i. Called Alex whose major is CS, GPA is 3.4, credit hours = 54, height is 170 cm, weight is 150 pounds and age is 18.
- b. Create an object of **class Doctor**
 - ii. Called jack whose specialty is “Dermatology”, years = 4, height is 173 cm, weight is 179 pounds and age is 40.
- c. Print out the student’s major, GPA, year and age.
- d. Print out the doctor’s specialty, height, weight and salary.