|  |
| --- |
| Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# CS 122 Lab 2

01/31/2017

## Task 1—Person class hierarchy

1. Implement a superclass Person. A person has a name and a year of birth (use instance variables to declare). Add the toString method to the superclass Person.

Recall that the toString method returns a string representation of the object. toString method is invoked automatically when an object is printed.

1. Make two subclasses, Student and Instructor, that inherit from Person. A student has a major and an instructor has a salary (new instance variable for each subclass). Implement these two subclasses as well.
2. Implement the constructors for subclasses. Use key word super to invoke the constructor of the superclass.
3. Implement the toString method for each of the subclasses. Use key word super to invoke the version of parent’s toString method.
4. The test program PersonTester.java is provided. You do not need to modify the test program. Run the test program and see if you get the expected output. If not, figure out the problem and generate the output as expected.
5. Write down the term of java technique you used to make each class have a toString method but they differ in detail:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Task 2—Abstract Class

1. Continue with the above program. Now you want all subclasses have a method that can return the average age of its object type, i.e., the average age of students and average age of instructors. Since Person has no idea what the average age is for each subclass, we need to define an **abstract** method in the Person class and leave the details for subclasses to implement.

Add an abstract method avgAge() to the Person class. Once you add an abstract method, the entire class should be abstract too. Hint: abstract method has regular header but no body.

1. Now any subclass of Person must have an avgAge method; do this to both Student and Instructor. This time, you should be able to define the body of this method since each subclass has its average age. Suppose the average age of students is 18 and average age of instructors is 41.
2. Recompile the driver program. You should get an error. The error message tells you exactly what the problem is. Fix this and recompile PersonTester.java.
3. Add code to the driver program to invoke the avgAge() method of Student and Instructor. Run the driver program and see the output.
4. Can you summarize using one sentence when do you need an abstract method?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Show the instructor when you get the above done.**