

METRO STATE UNIVERSITY

ICS 141 - 02: Problem solving with programming Spring 2023

Lab 2: Objects and classes

Friday, January, 20th, 2023

Total points: 20

NOTE: To receive credit for this lab assignment, you must:

- (1) attend the lab session, **AND**
- (2) demonstrate your solution to the instructor before you leave.

If you are not able to finish the work before you leave class today, upload an incomplete version before you leave and then upload a completed version by 11:59 PM on Friday, January, 20th, 2023.

Part 1. Declaring 'Dog' and 'DogDriver' classes

In this Lab, you will implement a Java class, called 'Dog', which can be used to represent dog objects in an application to be used, for example, in an animal hospital. You will then test your class by implementing another class, called 'DogDriver'.

To start, Open Eclipse and create a new project and name it 'DogApplication' then follow the following steps:

1. Create a class called 'Dog' inside the 'DogApplication' project.
2. Declare the following two instance variables in 'Dog'. Remember that instance variables must be declared as `private`.
 - a. **name** (of type `String`).
 - b. **age** (of type `int`).
3. Declare a constructor for the 'Dog' class that takes two input parameters and uses these input parameters to initialize the instance variables.
4. Now create another class in your project and call it 'DogDriver'. You will use this class to test the 'Dog' class. The 'DogDriver' class should include only the `main` method.
5. Inside the driver's `main` method, declare two variables of type 'Dog' and call them 'myDog' and 'yourDog', then, assign these two variables to two instances of 'Dog' with any values of your choice for name and age. (Note: you have to use `new` to instantiate the dog objects).
6. Go back to the 'Dog' class and implement the 'toString' method and then call this method from 'DogDriver' to print the dogs' information.

CHECKPOINT 1: Make sure the information about your dogs is printed correctly and show the output to the instructor.

Part 2. Adding more methods to the 'Dog' class.

In this part, you are going to add (and test) more functionality to your 'Dog' class.

7. Assume that, in the 'DogDriver' class, you tried to print only the names of the two dogs that you created in step #5 using the following statements:

```
System.out.println(myDog.name);           System.out.println(yourDog.name);
```

Write down the error message you get when you type these statements?

8. In order to be able to print only the dog name, go back to the 'Dog' class and implement a **getter** method for the dog's name instance variable.
9. In the DogDriver class, print the dogs' names.
10. Assume now you want to change the age of 'myDog' using the following statement:

```
myDog.age = 20;
```
11. **The previous statement will not work.** You must implement a setAge method in order to be able to change the dog's age. So, go back to the 'Dog' class and implement the setAge method. Then, call the setAge method from the 'DogDriver' class two times to change the age of both dogs.
12. Call the toString() method to print the dogs' information again to make sure that the ages are correctly changed.
13. Draw the UML diagram of the Dog class.

CHECKPOINT 2: Show the output of your completed program and your UML to either the instructor.

Part 3. Upload your work to D2L

1. Right click on the src folder and choose the Export... option.
2. When asked to choose an Export Wizard, double click on General.

3. Then click Archive File and Next. Make sure the radio button next to Save in zip format is selected.
4. Use the Browse function to navigate to the H: drive, then make up a name for your file. You can name your archive file anything but it helps me find your work if you use your last name and the name of the lab so choose something like GhanemLab03. Click Finish.
5. Now login to D2L. From the home page, click on the Assessments drop down menu and select Assignments.
6. From the Assignment Submission Folders page, select the Lab3 drop box.
7. Click on Add a File, browse for the archive (.zip) file you created with Eclipse and Upload it. You can add a note if you like but you don't have to. Click Submit to finish up.
8. Draw the UML on a word document and upload that document to the Lab 3 drop box on D2L.

NOTE: To receive credit for this lab assignment, demonstrate your solution (even if it is partially completed) to the instructor before you leave.