

## Lab 13: Creating Instantiable Class Lab

### Learning Outcomes

By the end of this lab:

- Write code to initialize instance variables.
- Describe the difference between class and instance variables.
- Write code for an instantiable class.
- Testing your instantiable class with a second class known as a *driver* or *test* class.
- Commenting your instantiable class.



### Exercise A: Trace & Explain

With your partner, trace and explain each of the following (also found in [Trace Explain.txt](#)). Check your understanding using Java Visualizer or IDE.

1. In the following class:
  - a. If **CODE** is replaced with **count**, what value gets printed and why?
  - b. What should **CODE** be to print out the **count** field of a Bug object?
  - c. What should **STATEMENT** be replaced with to assign the parameter to the field?

```
class Bug {  
    private int count = 0;  
    void methodA(int count) {  
        System.out.print( CODE );  
        STATEMENT  
    }  
    public static void main(String[] args) {  
        Bug b = new Bug();  
        b.methodA(5);  
    }  
}
```

2. Is it possible for two different Potato objects to have different values in the size field?

```
public class Potato {  
    // static = class variable  
    static int size;  
}  
  
- main  
    Potato p1 = new Potato();  
    p1.size = 5;  
    System.out.println(p1.size)  
    Potato p2 = new Potato();
```

```
System.out.println(p2.size);
```

No, since it is a static variable

3. Is it possible for two different Onion objects to have different sizes?

```
class Onion {  
    // both instance variables  
    String brand;  
    int size;  
} // because it is not a static variable
```



## Exercise B: Create a Class

Given the following declaration and constructor call:

```
Flower f;  
f = new Flower( "daisy", 4.0);
```

Write the Flower class and its constructor in which "daisy" is the description and 4.0 is the height in inches.

```
Class Flower {  
    // two instance variables, for description and height  
    String description;  
    Double height = 0.0;  
    Flower(String description, double height) {  
        This.description = description;  
        This.height = height;  
        //System.out.print(this.description);  
        //System.out.print(this.height);  
    }  
    public static void main(String[] args) {  
        Flower f;  
        f = new Flower("daisy", 4.0);  
        //f.methodA( "daisy", 4.0);  
    }  
}
```



## Exercise C: Write and Instantiate Classes

Work through this [tutorial for writing and instantiating classes](#) and using the javadoc tool to create the JavaDoc API for them. Let your TA know how far you get and what questions you have.

Task1:

Candidate's name: string; Political party: boolean; Office: integer

Task2:

Yes, we need; we should get the value and set the value in each method

Task3:

Parameter: such as a algebra in the function which can be setted to oher value;

Instance variable: holds the value and can be called through the class;

Local variable: declared in methods,constructors, or blocks

Task4:

Task 5:

Task 6:

## Exercise TA: Demonstration and Discussion

*This exercise is utilized to determine 3 points of the lab grade. We suggest reviewing these before you discuss these with your TA.*

1. Exercise A: Trace and Explain
2. Exercise B: Create a Class

3. Exercise C: Writing and Instantiate Classes and using the javadoc tool. Show your TA how far you got and be prepared to discuss the questions in the tutorial.



## Exercise D: Shortest Code

What is the shortest code you can write to demonstrate the answer to this question?

Create a class House with an instance method. Within the instance method, create a new object of class House. Within the same instance method, will it be possible to access the object's private fields?



## Exercise E: More Trace & Explain

With your partner, trace and explain each of the following (also found in [traceExplain2.txt](#)). Check your understanding using Java Visualizer or Eclipse.

- What will the following code display to the console when it is run?

```
public class Card {
    private final int id;
    private static int count = 0;
    public Card() {
        id = ++count;
    }
    public String toString() {
        return "" + id;
    }
    public static void main(String []args) {
        //Describe what the following statement does?
        //Will the Card() constructor be called?
        Card []cards = new Card[3];

        //How many id fields will be created?
        //How many count fields are there?
        for ( int i = 0; i < cards.length; i++)
            cards[i] = new Card();

        //Will this call the toString method of Card?
        //What will print out?
        System.out.println( cards[2]);
    }
}
```
- Is it possible for different Bag objects to have different numbers of Onions?

```
public class Bag {
    ArrayList<Onion> list = new ArrayList<>();
}
```

## Additional Learning Materials

When you have mastered everything in this lab, and in previous labs, then you are welcome to learn from additional learning resources available on the web and beyond this course:

<https://codingbat.com/java>

<https://www.khanacademy.org/computing/computer-programming>

<https://techdevguide.withgoogle.com/>

<https://code.org/>

<http://programmingbydoing.com/>

*Note: Team Lab is focused on pair programming and open discussion and so it is not appropriate to work on individual programming assignments.*

Faculty Associates contributed the Writing and Instantiating exercise.

Lab © 2018-2020 Jim Williams