Unit 2: Using Objects

Topic 1: Intro to Objects

Name:	

Rectangles!

0. Create a new IntelliJ project named LASTNAMEU2T1Lab1 (e.g. "KaufmanU2T1Lab1") and login to GitHub on IntelliJ.

1. Create a new class in the **src** folder named **`RectangleRunner`** and copy/paste **this code**.

2. Create a second class in the **src** folder named **`Rectangle`** and copy/paste **this code**.

3. The `RectangleRunner` class has a `main` method but the `Rectangle` class does not; why is that?

Note how there are no  on the left side of the `Rectangle` class!

4. Find the "constructor" method in the `Rectangle` class at **line 10**.

A. What do you notice about how it's named?

B. What do you think it does?

A.

B.

5. Jump back to the `RectangleRunner` class.

A. What keyword (in orange) is used to create two different `Rectangle` **objects** on lines 5 and 9? Where have we used this keyword before?

B. This code: `new Rectangle(5, 6)` calls the *constructor* method of the `Rectangle` class, passing 5 and 6 as "actual parameters" into the method. Look at the constructor again in the `Rectangle` class; what does the 5 represent? What does the 6 represent? How can you tell which is which?

A.

B.

6. Execute (run) the `RectangleRunner` class and view its output!

7. Lines 6 and 10 of `RectangleRunner` are examples of "*calling a method*" on a particular `Rectangle` **object**. Look at the output, then go to the `Rectangle` class and find the `printArea()` method. How does the method calculate the area?

8. In the `RectangleRunner` class, create another `Rectangle` object named `rect3`. Give it a length and width of your choosing. Then call the `printArea()` method on your new object. Run the code to test that it works!

Copy and paste the line(s) of code that you wrote below:

9. Try creating another `Rectangle` object, `rect4`, with a width of 6.5 and length of 10.5. What happens and why do you think that is? Hover your mouse over the red squiggly in IntelliJ and see if you can figure out the error!

Is it a syntax/compiler error, or a runtime error?

Once you have the problem figured out, go ahead and delete `rect4`.

10. Go into the `Rectangle` class and try writing a new method, `printPerimeter()`, which calculates and prints the perimeter of the `Rectangle`, similar to area. Start the method with "`public void`" like `printArea()` -- we will talk about what these words mean soon!

Then test your method by going to `RectangleRunner` and adding code to call the new method on each of the `Rectangle` objects (`rect1, rect2, rect3`). Confirm the output is what you expect!

Copy and paste the line(s) of the method you wrote below:

11. If you had to guess, what do you think `public` means? What about `void`?

Partner Compare

Take some time to **compare** your answers for **1-11** with your partner.

What is your partner's name?

Did you come up with similar answers?

Cats!

12. Create two new classes in the src folder: **`CatRunner`** and **`Cat`.**

Give the `CatRunner` class a `main` method, and for the `Cat` class, copy/paste **this code**.

Notice that you now have two classes in your project, `RectangleRunner` and `CatRunner`, which both have main methods, so both of them are executable:

13.
A. In your `CatRunner`'s `main` method, write code to create a `Cat` object named `cat1`. You will need to look at the constructor in the `Cat` class to determine what values should get passed in as parameters! Choose the parameter values for your cat.

B. Next, look into the `Cat` class and find two different methods. What are the two methods named?

C. Pick one of the methods then write code to call that method on `cat1`. Run the code to see the output.

D. Write code to call the *other* method on `cat1`, then run the code to see the output.

E. Write code to create a second `Cat` object, `cat2` (again choosing your own parameter values), and call both methods on `cat2`.

Copy and paste the line(s) of code that you wrote below:

14. What happens if you switch up the order of the parameters when you create your `Cat` objects? In other words, determine if both of these are "valid" calls to the `Cat` constructor (i.e. will both compile):

```
new Cat(5, "Fluffy", 8.5)
new Cat("Fluffy", 5, 8.5)
```

What happens when you mix them up and why?

15. Challenge! In the `Cat` class, modify the `introduce()` method so that it prints: "Hello my name is ___ and I am a younger cat" if `age` is less than 7, and "Hello my name is ___ and I am an older cat" if `age` is 7 or older.

Run your `CatRunner` again to test! You should see the updated output for both cats. Make sure one of the cat's ages is less than 7 and one is greater than 7, just to test the method!

Copy and paste your updated method below:

16. **Freestyle!** Play around with any of your four classes (`RectangleRunner`, `Rectangle`, `CatRunner`, `Cat`). Change things, add things, write some new code.

Or as an added challenge, try to create your own simple class from scratch!

Jot down one thing you tried and what you figured out! Feel free to copy/paste code if you are particularly happy with it :)

Done!

Submit in Google Classroom:

Turn in