Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**EGR 222 Java Intro, OOP Intro, static**

1. Refer to example code here:

<https://github.com/mikiehan/EGR222-SP19-Lecture01-Example-Code>

1. In Java everything is object. Java is highly object-oriented language.

2. In Java class name should be the same as file name that contains the class definition.

3. What is class? (Not specific to Java)

Short answer: It is a template for an object. Creates a type for an object

Some analogies:

1. blueprint for buildings
2. cookie cutter for cookies

Below are long answers:

4. Two key features of a class in OOP (Not specific to Java)

Data and Actions

Show example of Point

5. Specifically for java,

1. We call data fields : We used to call this member variables in C++
2. We call action methods : We used to call this member functions in C++

6. What is the difference between a function and a method?

Function – defined outside of a class (\*We call the function inside of class as member function in C++)

Method – defined inside of a class

(False) In Java, we can have a function.

No! Only methods exist in Java!

7. What is object?

Short answer: The actual product created by the class

8. Object has a fancy nickname: instance

Throughout the class, we will use object and instance interchangeably.

Show example of Point

9. When a class creates an instance/object, we call it object instantiation.

“Object is instantiated”

“We instantiated an object”

What a sophisticated word that make us look smart! Let’s start using this word ☺

10. (Almost always) a class itself cannot actually perform anything. In order to do something we need an instance. Java is so object-oriented it almost always require you to instantiate an object.

Unless… we use a keyword static!

Show example of Point

11. Java static keyword makes fields/methods belong to a class not to an instance.

12. static field: There is only one copy shared by all instances of the class.

Static field belongs to a class (not to a specific instance)

We don’t need any instance to refer the field!!

The opposite of static field (non-static field) is called instance field. We need an instance for this in order to use the instance field. That is why it is called **instance** field.

13. class constant: A constant (something that value does NOT change)

What keywords to use to declare a class constant

final – freezes the value upon initializing

static – to keep one copy and make all instances share. (Also we can use the constant without having to have an instance. Directly refer by className.constantName)

public/private – determines access level. public can be viewed/used by the outside classes. private can be viewed/used only within the same class

14. static method:

A method belonging to a class. We do NOT need an instance in order to use/call it

The opposite of static method (non-static method) is called an instance method

15. When to use static method?

When a method is not doing anything specific to an instance

(The opposite of static method is an instance method. You need an instance to call the instance method. That is why it is called **instance** method)

\*\* IT IS ALWAYS THE BEST to make a method static when you are not doing anything with the instance fields

\*\* When do you have to use instance method? When the method is doing something with the instance fields.

16. One of syntax differences between Java and C++.

Instead of cout: System.out.println

I highly recommend referring to Java document API.

For example, if you want to look something up regarding Java String <https://docs.oracle.com/javase/10/docs/api/java/lang/String.html>