Tophat Quiz

You will have 5 minutes to answer two quesions

Class Terminology

- The **class** construct defines a new type that can group data and methods to form an **object**
- Class **public member methods** indicate all operations a class user can perform on the object
- The **new** operator explicitly allocates an object of the specified class type
- The **"." (dot)** operator, known as the **member access operator**, is used to invoke a method on an object

Class Terminology (cont.)

- A class can contain private or public variables, known as **fields**
- Both class variables and methods are collectively known as **class members**

Example of a Class:

Access Modifiers: Public vs Private

- **Private**: A variable or method that is private can only be accessed from within the class that it was declared (Most restrictive)
- **Package**: Can only be accessed by another class from within the same directory (package) as the class with this variable or method is declared (Second most restrictive)
- **Protected**: Don’t worry about this one just yet
- **Public**: Can be accessed from any class (least restrictive)

Let’s apply this knowledge...

More Class Terminology...

- **Abstraction** means to have a user interact with an item at a high-level, with lower-level internal details hidden from the user
- **Encapsulation** is defined as the wrapping up of data and code (variables and methods) under a single unit (a class)
    - It does this by hiding the “nitty gritty” code and variables from the user, and providing a simple **abstraction** for the user, this is known as **information hiding**

Mutators and Accessors

- Commonly known as: **Getters** and **Setters**.
- These are methods that allow access to hidden data

Let’s look at an example...

Private Helper Methods

- A method that has a private access modifier.
- The type of method usually is used to help a public method carry out some task

Let’s look at an example...

Constructors

- **Constructors** are a block of code with in a class that only gets called when an object gets initially created (i.e. it only gets called once)
- A constructors signature looks very similar to a method's with the exception of lacking a return type, and it must share the same name as the class
    - A constructor can even define parameters / take in input arguments
- The purpose of a constructor is to put an object into an initial state, usually this only means initializing variables

Let's apply this knowledge...

More Class Terminology...

- A **default constructor** , is a constructor with no input arguments.
    - If you do not specify a constructor in your class, java will conveniently create a default constructor for you, without you even realizing it
- **Constructor Overloading** refers to creating multiple constructors, each with different input argument types

Let’s look at an example...

Objects and References

- A **reference** is a variable type that refers to an object, it can be thought of as storing the memory address of an object
- The **new** operator allocates memory for an object, then returns a reference to the object's location in memory

Primitive Types vs Reference Types

- A **primitive type** variable directly stores the data for that variable type, such as int, double, or char. Ex: int numStudents = 20; declares an int that directly stores the data 20.
- A **reference type** variable can refer to an instance of a class, also known as an object, via the object's memory address

Pass by Value

- When assigning a variable to another variable, the value at the memory location of the first variable, get **copied** to the memory location of the second variable
- This also occurs when passing a variable as an input to a method
- What happens when we pass an object as an input to a method? Does it get copied into the method parameter?