1. What are the two categories of data in Java?

Primitive data types (int, double, boolean, etc.)

Objects (Scanner, String, BufferedReader, etc.)

2. What is a class?

A class is a blueprint for an object, it contains the data and functionality that define what the object is, how it works, and what it can do.

3. What is an object?

Objects are the things constructed by following the class blueprint. They are also called instances of the class.

4. What are the 3 main parts of a class?

Fields: the data values that are stored in the class/object

Constructors: the code that is run when an object is created from the class

Methods: the code that can be called from the class/object

5. Write out the Java syntax for a class header

<access\_modifier> class <Name>{

6. Which of the following are invalid class names? Identify the problems with the invalid names

Rectangle

**circle** // Class names must be capitalized

**Small Square** // Class names cannot have a space

Large\_square // Allowed but not encouraged, this is Python’s naming convention

MediumSquare

**3DBox** // Class names cannot start with a number

Box3D

**String** // String class already exists

7. What is a constructor that has no arguments called?

A no-args constructor/default constructor

8. Write out the Java syntax for a no-args constructor

<access\_modifier> <Class\_Name>(){

9. How do you set the default values in a class?

To set the default values in a class, you can set the values in the field declarations (right after the class header), or you can set the values in the default constructor

10. What does the this keyword do?

The this keyword allows you to differentiate between class-level scope and method-level scope if two variables have the same name. For example, if there’s a class variable named length, and I declare a second variable named length in one of the class methods, any reference to length inside the method will refer to the method-level variable, and this.length will refer to the class-level variable.

11. What does the static keyword mean when applied to a...

**variable:**

the variable refers to the class in general, not any specific instance of the class. For example, all cars have 4 wheels so that variable would be considered static

**method:**

the method can be called without creating an instance of the class. Instead, the method is called using the name of the class. For example, Integer.parseInt() is a static method, since it can be called from the Integer class without creating a new Integer.

12. What is a static reference?

A static reference is any reference to a class using the class name itself. For example, if you create an instance of the Integer class by typing

Integer i = 10;

any references to i would be a non-static reference. Rather, any references to Integer itself, such as

Integer.parseInt()

would be static references.

Additionally, if you see the error

Cannot make a static reference to a non-static field

this means that you are referencing an instance variable in a static method. You need to create an instance of the class it is in before you can use that variable.

13. What are getters and setters?

Getters and setters allow an objects data to be referenced in a safe/restricted way, such that the fields cannot be modified directly. Instead of changing an object’s height, for example, instead of typing

object.height = newHeight

we would call the setter by typing

object.setHeight(newHeight);

14. Why is it helpful to make a class’s fields private and use getters and setters instead of allowing the fields to be manipulated directly?

Using setters, we can control what data is stored in the object’s variables. For example, we can prevent negative values from being set to a variable that should only ever be positive, and we can make sure other related pieces of data are also updated if necessary, like if we change the height of a rectangle, the area and perimeter should also be updated.