

# Object-Oriented Programming

DSC 30 Summer 2023

## Scavenger Hunt

How is object-oriented programming different from data-oriented programming (like Pandas)?

What does “static” mean?

What does “null” mean? How is it different from the “void” return type that we learned previously?

What is the purpose of an interface?

Why do we use private fields? Is it the client or the implementer who can access private fields?

# Burning Questions

At the end of class, [submit your burning questions](#). Your participation is required; this and/or the integrity of scholarship agreement form will be used to determine commencement of academic activity.

One question is like the differences between data oriented programming and object oriented programming. I am still confused about these two.

How is object-oriented programming and data structures related to each other? Why do we need to know object-oriented programming before diving into data structures?

- In what scenarios is it more useful to use Java instead of Python?
- The object oriented programming section is inspiring

What's client vs implementer of the code? Which part of code for each? Still not sure the difference

What's a field? difference between field and a class?

I don't quite understand the double [int] CompareTo method and interface part mentioned in today's class. Maybe I need to read the textbook and go to office hours later.

- I'm still generally confused on interfaces,
- One question I have is when interfaces vs. classes would want to be implemented.

Come chat with us! I think it would help to clarify what part of this question is confusing! I will send an email to everyone in the class with the office hours Zoom meeting right after this section.

## Are keywords like void and static absolutely necessary? What happens if we leave them out? When do we use them?

- Also I am still a bit confused about void versus null return types because don't they both return nothing?
  - Is the only difference between null and void that null is a placeholder while [void] is literally nothing. If that's the case, why is null being a placeholder important? How is it important in data science?
- When should a method or variable be static?
  - When it is shared among all instances of the class. For example, if you have a static variable 'name' in the class Dog, all the Dogs you create will share the same 'name'. When you change the 'name' of one Dog, it will affect all Dogs.
- what would happen to your code if you had a private static method? Would it have any effect on the code?
- Still feel confused about the difference between 'with static' and 'without static'.

Static: Attached or associated with the entire class.

Void: Means that it's nothing returned by a function.

Null: Like a placeholder for a future object. Python has something called None.

## Are public/private similar to something in python? In python, we could directly access some variable in a function, or we can define some methods like "getValue" which returns the value of that function. Are those similar to public/private in Java?

- I was thinking that financial field could use the private fields for their clients' sakes. I was wondering what could be an example of a financial field company using a private field using java.
- What situations would use a getter method in order to provide access to a private field? If private fields are meant to restrict access to certain fields, then wouldn't you just make that field public instead if you're going to be providing it to a client anyway? Or do we have the getter method in order to only give certain clients private information?
- What is the purpose of "private field" if it makes accessing instances more complicated?
- I think that I am not familiar with JAVA yet, I still don't understand what is interface doing and why we are asked to use private field
- Another question I have is what the benefits are of making a field private if they can't be accessed from anywhere

Professor talked about how we do not have to define this unlike how we define self in `__init__` definition. I was wondering how the java would know we are going to use the "this" method without actually defining it in the code.

This is a special feature of Java: even though it's not written, it's provided automatically by the Java programming language.

Can all datatypes in Java be an array of objects? (e.g. `int[]`, `String[]`)

Yes, even things like `Account[]`. It is a little tricky to create arrays of "generic types".

At the end of the lecture, we did learn how lists can sometimes be more useful than using arrays. So when is an array more useful than a list? and why do we prefer using arrays for OOP?

An array is preferred over a list if you know you need a precise capacity or length for the list, and don't need to change that length or capacity later.

How many parameters can a function have before Java crashes?

There's a limit of 255 parameters.

Do you have any idea or better way of naming objects that is efficient and easy to recall?

Naming is a style convention that can vary by organizational and personal preferences. I don't think we are using a strict style guide for this class, but here are some general guidelines:

- Your variable/class/function/etc. names should be relevant to their function (e.g. you should be able to get an idea of the role of a variable in your code just by looking at it)
- Generally, we use lower case, camel casing for variables in Java code. For example, we would tend to name a variable `nameLength` (camel casing) instead of `name_length` (snake casing, as is standard in Python).
- There are many other "standard" practices for capitalization for naming classes vs. functions, etc. The general goal is to be able to identify what kind of item a name refers to based on how the name looks.

More specifically, here are some resources that you can check out for this and other style questions: [Oracle Code Conventions](#), [Google's Java Style Guide](#), [DSC30 Fall 2018 Style Guide](#)

## Can we have multiple classes in one java file?

Yes, but only one class in the file can be public (as you saw in the Java Tutor).

## What is the <E> or <Account> stuff that we've been seeing?

When we are implementing a class, sometimes we don't know or don't need to know what the type of data the class will be working with or holding. Sometimes we also want to leave the liberty to the client (users).

<E> is a generic used for indicating the liberty

If the class is working with Account class objects, then you use <Account> whenever the type needs to be specified.

For example,

```
ArrayList<Account> arrList = new ArrayList<Account>();
```

Will create a new ArrayList holding Account objects.