

Write on whiteboard

Three OOP principles

- **Encapsulation:** the concept of hiding values or state of data within a class, limiting the points of access
- **Polymorphism:** the ability for our code to take on different forms
- **Inheritance:** the practice of creating a hierarchy for classes in which descendants obtain the attributes and behaviors from other classes

Java

Module 1 - Day 9

OOP Basics &
Encapsulation

Today's Objectives

- **Encapsulation**
 - How is it implemented?
 - Why is it used?
- **Access modifiers:** public vs private
- Define **loosely coupled**
- **Static** and **Final**

The Three** Basic Principles of OOP

Three OOP principles

- **Encapsulation:** the concept of hiding values or state of data within a class, limiting the points of access
- **Polymorphism:** the ability for our code to take on different forms
- **Inheritance:** the practice of creating a hierarchy for classes in which descendants obtain the attributes and behaviors from other classes

... or ...

Just reassure yourself that this OOP is easy as P.I.E.

**or A PIE if we include Abstraction... more on that later



Let's Design Some Classes!



Encapsulation

- The packaging of data and functions into a single component (e.g. a class with state/properties and behavior/methods)
- Allows us to hide the implementation details of a class to help ensure consistent state and to facilitate keeping our classes loosely coupled
- Goals:
 - ◆ Makes code extendable
 - ◆ Makes code maintainable

Static Variables (Class Variables)

→ Static Variables

- ◆ A variable that are common to all objects of a class.
- ◆ Fields that have the static modifier in their declaration are called static fields or class variables.
- ◆ Associated with the class, not an object.
- ◆ Every instance of the class shares a class variable, which is in one fixed location in memory.
- ◆ Any object can change the value of a class variable, but class variables can also be manipulated without creating an instance of the class.

Constants (static final variables)

- The static modifier, in combination with the final modifier, is used to define constants.
- The final modifier indicates that the value of this field cannot change.

```
static final double PI = 3.141592653589793;
```


Static Methods

- Static methods are invoked with the class name, without the need for creating an instance of the class
- The Math library is a common and widely-used example.

What questions do you have?