



Java

Module 1 - Day 9

Object Oriented Programming (OOP) Basics:
Encapsulation

Module 1 Day 9 : Encapsulation and OOP

Can You ... ?

1. Define encapsulation, give a good example of it, how it's implemented, and describe why it's used.
2. Define "loosely coupled" and explain the characteristics of a loosely coupled system.
3. Define and use static methods and be able to describe what they're for.
4. Explain access modifiers and when you would use private versus public.
5. Define and use overloading in an OOP language.

The Three** Basic Principles of OOP

Three OOP principles

- **Encapsulation:** the concept of hiding the state of data within a class, and limiting the points of access for critical internal operations.
- **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 their parent classes

Dont panic...

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

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



Encapsulation

- **IS:** The packaging of data and functions into a single component (such as a class with state, properties, and behavior, methods)
- **ALLOWS:** Protection of the implementation details of a class and control over access to critical state variables and internal methods
- **BENEFITS:**
 - Makes code extendable by organizing it into classes
 - Makes code more easily maintainable
 - Helps ensure consistent state and facilitates loose coupling
 - ... among other things. What other benefits can you think of?

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.

Let's sharpen our
understanding of
classes!



Let's Design Some Classes!

