Intro Object-Oriented Programming

Lecture Flow

What is Object Oriented Programming?

The Four Pillars of OOP

Classes and Objects

Inheritance and Polymorphism

Encapsulation and Abstraction

Benefits of Object Oriented Programming

What is Object Oriented Programming?

Object Oriented Programming (OOP) is a programming paradigm that is centered around the concept of objects.

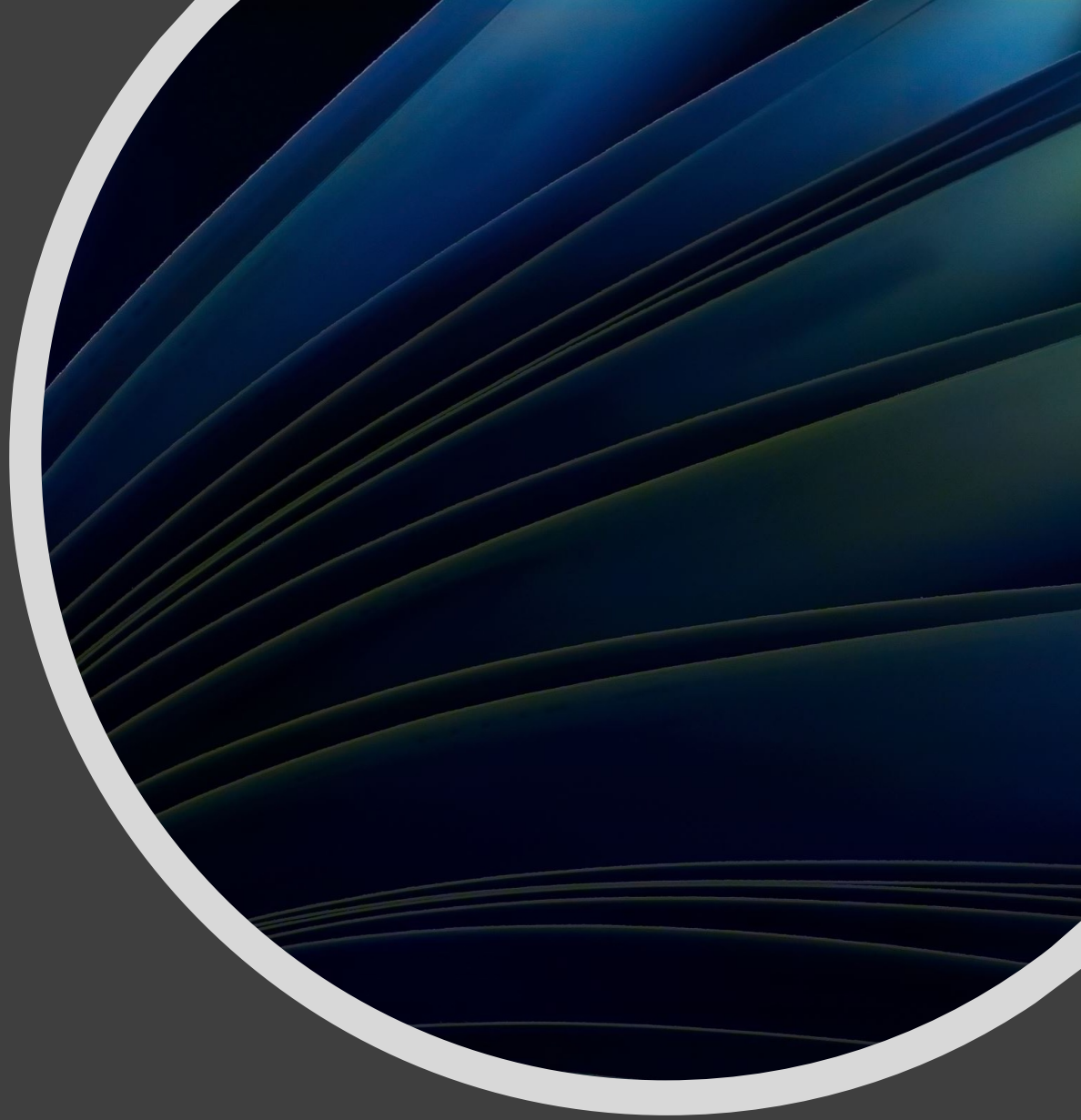
An object is an instance of a class, which is a blueprint for creating objects.

OOP allows developers to organize code in a way that is more intuitive and easier to maintain.

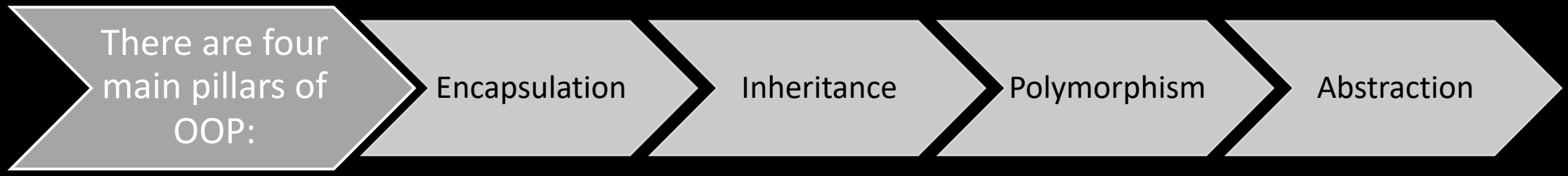
What is Object Oriented Programming?

In OOP, each object has its own state and behavior. The state of an object is defined by its attributes, while its behavior is defined by its methods.

This makes it possible to model real-world entities in code, as objects can be used to represent things like people, animals, or even abstract concepts.



The Four Pillars of OOP



The Four Pillars of OOP

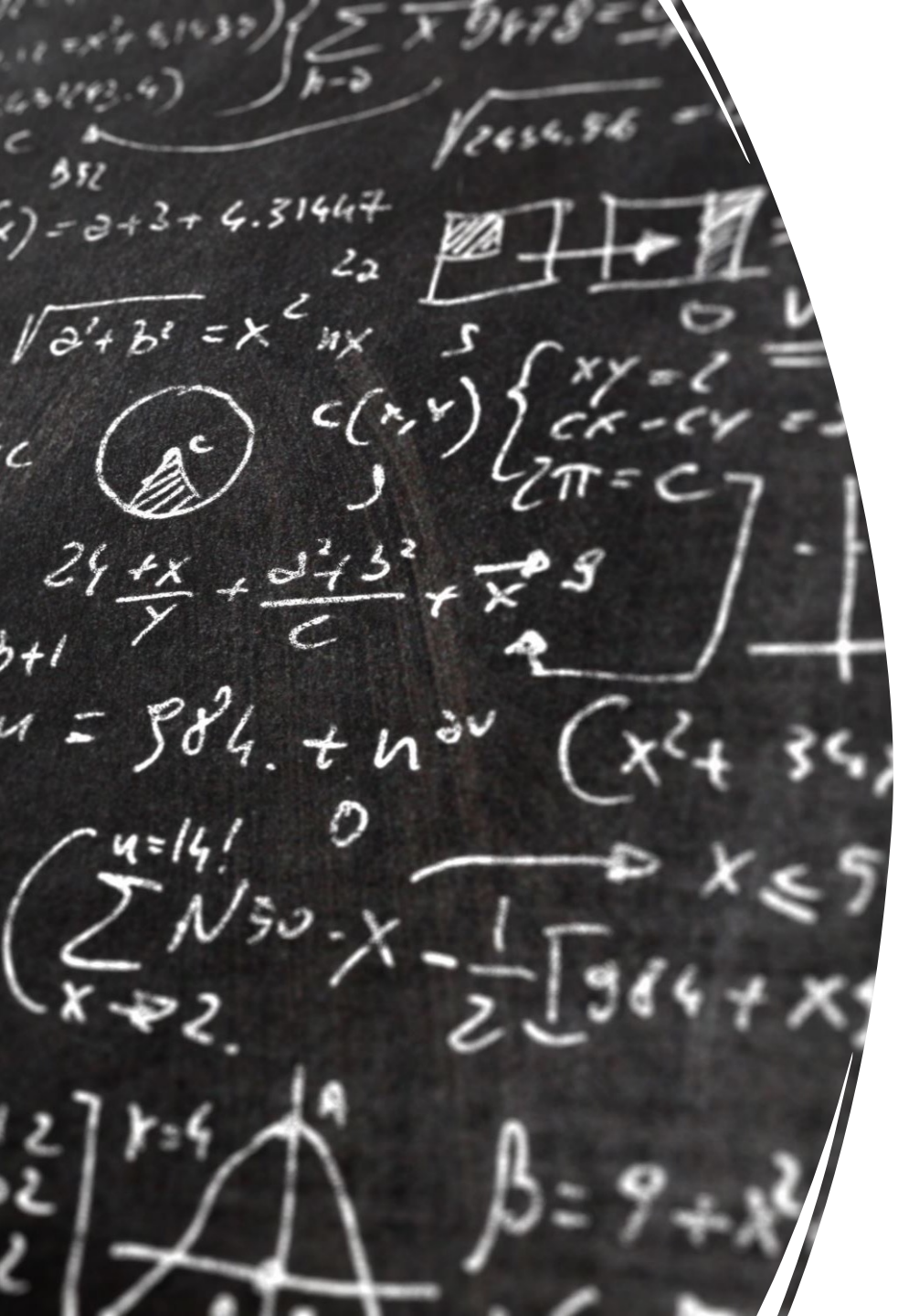
- Encapsulation refers to the practice of hiding implementation details from the user, and only exposing a public interface.
- Inheritance allows one class to inherit properties and methods from another class.
- Polymorphism allows objects to take on multiple forms, depending on the context in which they are used.
- Abstraction refers to the practice of modeling complex systems by breaking them down into simpler, more manageable parts.

Classes and Objects

In OOP, classes are used to define the structure and behavior of objects.

A class is essentially a blueprint for creating objects. It defines the properties (or attributes) that each object will have, as well as the methods (or behaviors) that each object can perform.

Once a class has been defined, objects can be created from it using the 'new' keyword.



Classes and Objects

- Objects are instances of a class. Each object has its own state and behavior, which is defined by the class from which it was created.
- Objects can interact with one another, and with the outside world, by calling methods on themselves or other objects.

Inheritance and Polymorphism

- Inheritance is a key feature of OOP that allows one class to inherit properties and methods from another class.
- This can help to reduce code duplication, as common functionality can be defined in a parent class and inherited by child classes.
- Child classes can then add their own unique functionality, or override existing functionality if needed.

Inheritance and Polymorphism

- Polymorphism is another important feature of OOP that allows objects to take on multiple forms.
- This can be achieved through method overriding, where a child class provides its own implementation of a method that was inherited from a parent class.
- Polymorphism can also be achieved through method overloading, where a class provides multiple methods with the same name but different parameters.

Encapsulation and Abstraction

ENCAPSULATION IS THE PRACTICE OF HIDING IMPLEMENTATION DETAILS FROM THE USER, AND ONLY EXPOSING A PUBLIC INTERFACE.

THIS HELPS TO PREVENT USERS FROM ACCIDENTALLY MODIFYING INTERNAL STATE, AND MAKES IT EASIER TO MAINTAIN CODE OVER TIME.

ENCAPSULATION CAN BE ACHIEVED THROUGH THE USE OF ACCESS MODIFIERS, WHICH CONTROL THE VISIBILITY OF CLASS MEMBERS.

Encapsulation and Abstraction

- Abstraction is the practice of modeling complex systems by breaking them down into simpler, more manageable parts.
- This can be achieved through the use of abstract classes and interfaces, which define a set of common properties and methods that can be shared by multiple classes.
- Abstraction helps to reduce complexity and increase modularity, making it easier to maintain and update code over time.

Benefits of Object Oriented Programming

- Object Oriented Programming offers several benefits over other programming paradigms.
- One of the biggest advantages is that it allows developers to write code that is more modular, flexible, and scalable.
- This makes it easier to maintain and update code over time, as changes can be made to individual components without affecting the entire system.

Benefits of Object Oriented Programming

- OOP also promotes code reuse, as common functionality can be defined in a parent class and inherited by child classes.
- This reduces code duplication and makes it easier to write clean, efficient code.
- Additionally, OOP makes it easier to model real-world entities in code, as objects can be used to represent things like people, animals, or even abstract concepts.

What's next?

Setup GitHub account and download GitHub Desktop

Plan first ”Mock” Sprint with Group due next week

Complete Lab 1