# Key Concepts and Notes

## Inheritance

Understanding how derived classes inherit properties and methods from base classes.

## Member Variables

Variables defined within a class to hold the state of an object.

## Main Object

The primary instance of a class that you interact with in your program.

# Copy Constructor

## Dereferencing a Pointer

When you dereference a pointer that points to an object, and that object is initialized by another object, the copy constructor is called.

## Passing an Object to a Function

When you pass an object to a function, and the function expects an object (not a reference or pointer), the copy constructor of the class that creates the object is called.

# Object Lifecycle in Functions

## Object Creation and Destruction

When a function is executed, all objects created within the function (including parameters passed by value) are destructed at the end of the function's execution.  
Note: Pointers not declared on the heap are not destructed this way.

# Destructors

## Non-Virtual Destructor in Base Class

If the base class destructor is not marked with the `virtual` keyword, derived class objects will not be properly deleted when using base class pointers.

## Virtual Destructor Example

Ensures that the destructor of the derived class is called when an object is deleted through a base class pointer.

# Additional Concepts

## Smart Pointer

A wrapper around a raw pointer that manages the lifetime of the object it points to, ensuring proper cleanup.

## Composition, Aggregation, and Association

- \*\*Composition\*\*: A strong relationship where the contained object cannot exist independently of the container object.  
- \*\*Aggregation\*\*: A weaker relationship where the contained object can exist independently of the container object.  
- \*\*Association\*\*: A general relationship where one object uses or interacts with another object.