

**Name : Aavani Rajesh Perumbessi**

**Roll no : 150096724059**

**Cohort : Mark Zuckerberg**

# What is Polymorphism?

## **One Interface, Multiple Forms**

Polymorphism allows objects of different classes to be treated as objects of a common base class.

## **Flexibility and Reusability**

By defining common interfaces, we can write code that works with various types of objects, promoting code reuse and adaptability.

# Static vs Dynamic Polymorphism

01

## Static Polymorphism

Resolved at compile time, based on the object's type. Examples include function overloading and operator overloading.

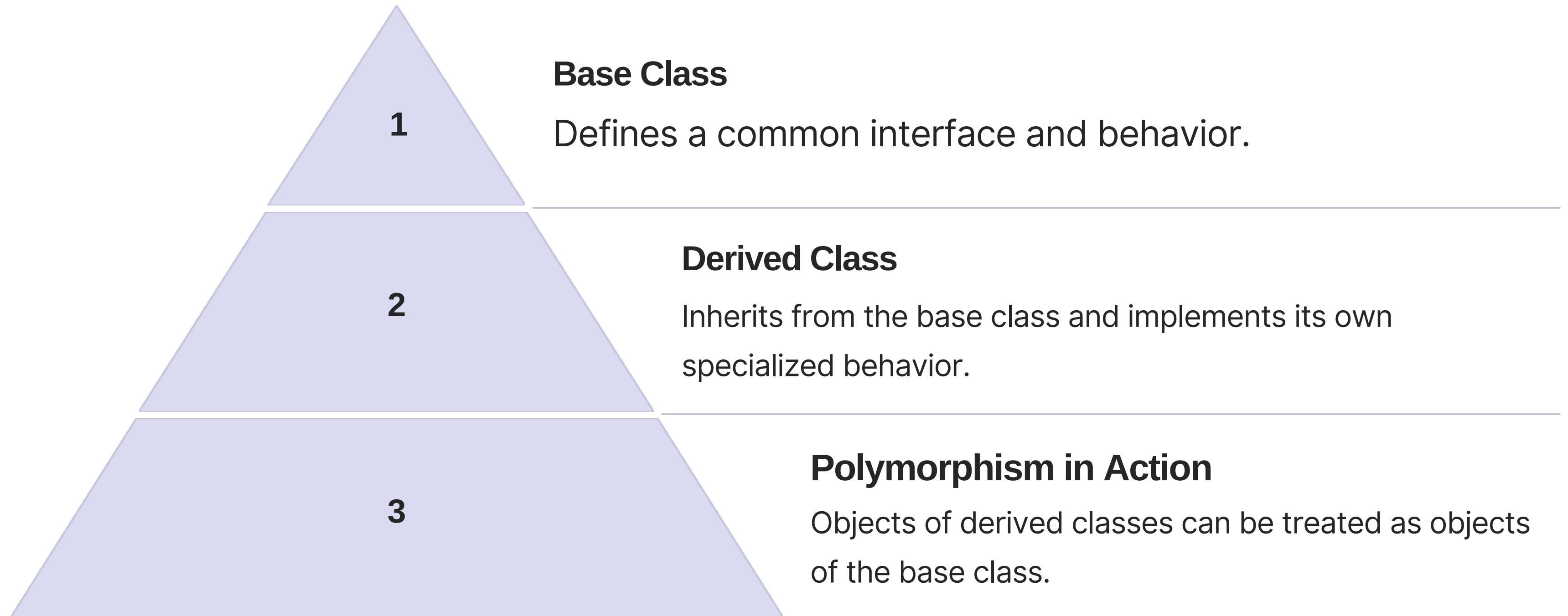
---

02

## Dynamic Polymorphism

Resolved at runtime, based on the actual object being used. This is achieved through inheritance and virtual functions.

# Inheritance and Polymorphism



# Overriding Virtual Functions

1

## Virtual Functions

Declared with the `virtual` keyword in the base class.

---

2

## Overriding

Derived classes redefine virtual functions, providing their own implementation.

---

3

## Runtime Behavior

The appropriate function is called based on the actual object type at runtime.

# Polymorphic Containers

## Flexibility and Efficiency

Use containers like vectors or lists to store objects of different derived classes.

## Unified Interface

Access and manipulate objects through the base class interface, regardless of their actual type.

## Example

A vector containing both `Dog` and `Cat` objects, accessed through a common `Animal` interface.

# Dynamic Binding and Late Binding

1

## Dynamic Binding

The process of resolving which function to call at runtime, based on the actual object type.

2

## Late Binding

Function binding is deferred until runtime, allowing for flexibility and dynamic behavior.

3

## Key Concept

Dynamic binding is the core mechanism that enables polymorphism in C++.

# Advantages of Polymorphism

- 1 Code Reusability**  
Write code that works with various types of objects.
- 2 Maintainability**  
Easier to modify and extend the code without affecting other parts.
- 3 Flexibility**  
Adapts to changes in the system without significant code alterations.





# Conclusion

Polymorphism is a powerful concept that allows for code flexibility, reusability, and maintainability. Understanding inheritance, virtual functions, and dynamic binding is crucial for leveraging polymorphism in C++. Embrace polymorphism to enhance your programming skills and build robust, adaptable software.

**THANK YOU!**