Object-Oriented Programming: A Guide

Object-oriented programming is a popular programming paradigm that uses objects to represent data and code in a structured way.

K **by Kawaljeet Kaur**

Principles of OOP

Encapsulation

Hides data and methods within an object, protecting it from external access. Encapsulation promotes data security and code maintainability.

Inheritance

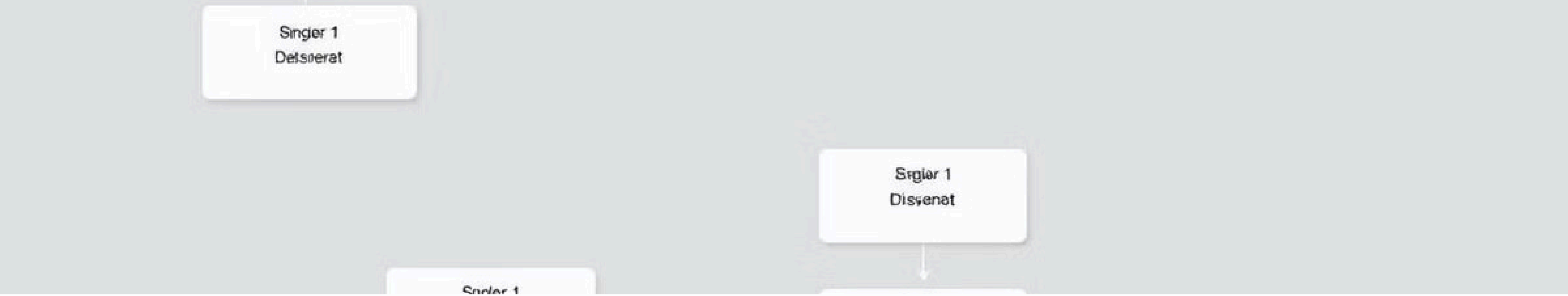
Allows classes to inherit properties and methods from parent classes, promoting code reusability and reducing redundancy.

Polymorphism

Enables objects of different classes to be treated similarly, allowing for flexibility and adaptability in code.

Abstraction

Simplifies complex code by exposing only essential features, hiding implementation details and improving code readability.



Classes and Objects

Classes

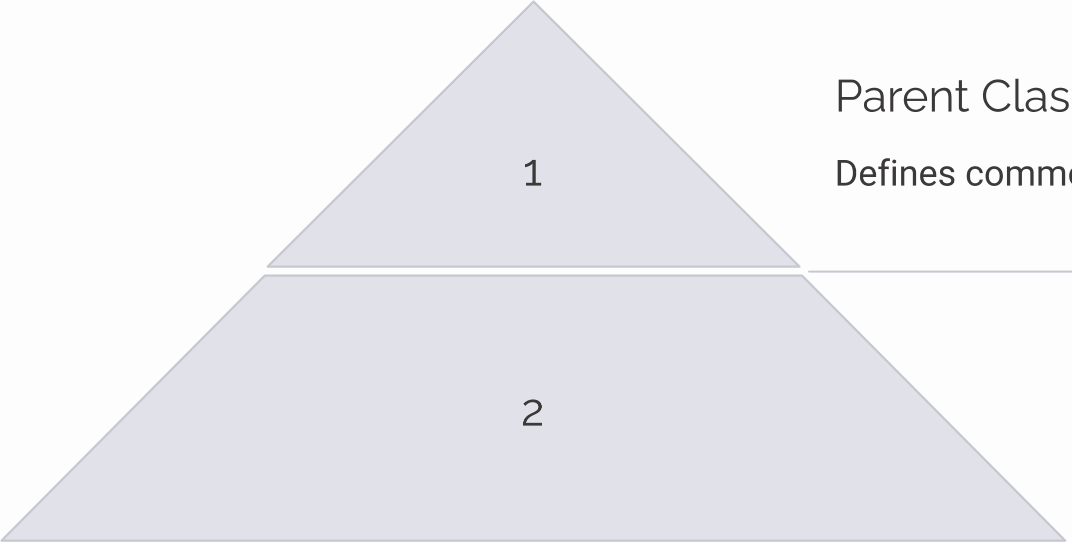
Blueprints for creating objects. They define attributes (data) and methods (behavior) for instances of that class.

Objects

Instances of classes, representing real-world entities. Each object has unique values for its attributes.

Made with Gamma

Inheritance: Building on Existing Code



Parent Class

Defines common attributes and methods for child classes.

Child Class

Inherits properties and methods from the parent class, adding its own unique functionalities.

Encapsulation: Protecting Data

Data Hiding

Internal data is accessible only through predefined methods, ensuring data integrity and consistency.

Controlled Access

Prevents direct manipulation of data, promoting controlled modification and safeguarding data from accidental corruption.

Polymorphism: Many Forms, One Interface

1

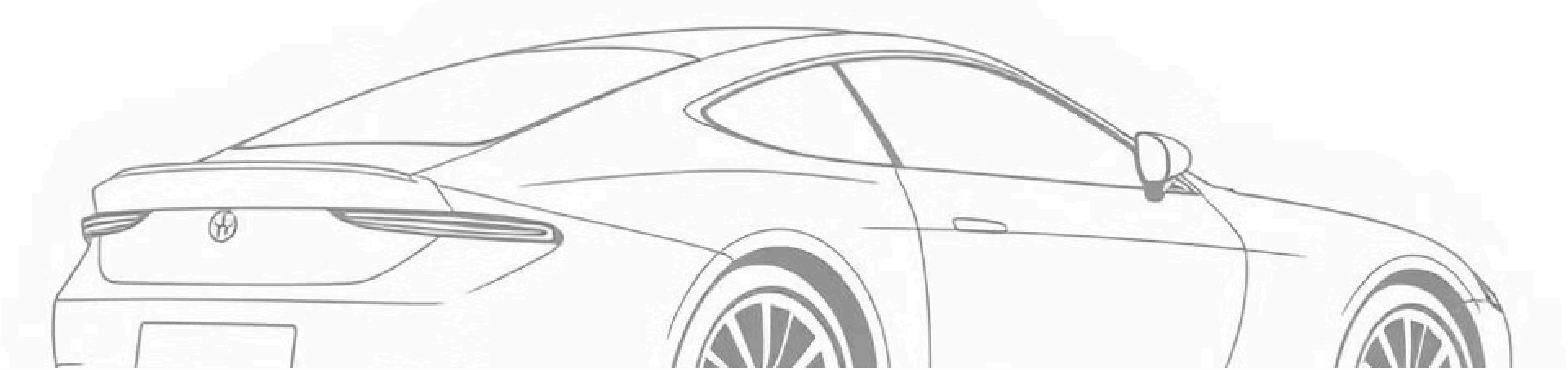
Method Overloading

Multiple methods with the same name, but different parameters, enabling different actions based on the arguments provided.

2

Method Overriding

Child class redefines methods inherited from the parent class, allowing customization of behavior.



Abstraction: Simplifying Complexity

□ Focus on Essentials

Hides implementation details, presenting a simplified interface to users.

□ Improved Readability

Reduces complexity and enhances code clarity by focusing on the essential functionalities.

Benefits of OOP

1

Code Reusability

Inheritance allows reusing existing code, saving time and effort.

2

Modularity

Code is broken into smaller, manageable components, improving maintainability.

3

Flexibility

Polymorphism enables code to adapt to changes without major revisions.

Conclusion

Object-oriented programming is a powerful paradigm that fosters code organization, reusability, and maintainability. Its principles empower developers to build robust and flexible software solutions.