HyperionDev

Welcome to this session:
Object Oriented Programming

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators answering questions.

HyperionDev

What is Safeguarding?

Safeguarding refers to actions and measures aimed at protecting the human rights of adults, particularly vulnerable individuals, from abuse, neglect, and harm.

To report a safeguarding concern reach out to us via email:

safeguarding@hyperiondev.com

HyperionDev

Live Lecture Housekeeping:

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- No question is daft or silly - ask them!
- For all non-academic questions, please submit a query: www.hyperiondev.com/support
- To report a safeguarding concern reach out to us via email: safeguarding@hyperiondev.com
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Stay Safe Series:

Mastering Online Safety One Week/step at a Time

While the digital world can be a wonderful place to make education and learning accessible to all, it is unfortunately also a space where harmful threats like online radicalisation, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

As a component of this BootCamp the *Stay Safe Series* will/is designed to guide you through essential measures in order to protect yourself & your community from online dangers, whether they target your privacy, personal information or even attempt to manipulate your beliefs.

Securing Your Mobile Device

- Use Strong Authentication
- Keep Your Device Updated
- Be Cautious with App Permissions
- Avoid Public & Unsecured Wi-Fi
- Install Security Software

HyperionDev

OOP

Learning Outcomes

- Explain what Object-Oriented Programming is and why it is used.
- Describe the four pillars of OOP: Encapsulation, Abstraction, Inheritance, and Polymorphism.
- Understand how OOP differs from procedural programming.
- Recognise real-world applications of OOP

Software Engineering

How would you describe the difference between a blueprint and the actual object created from it?

HyperionDev

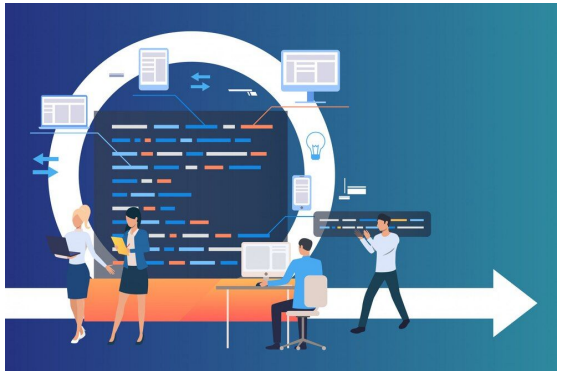
What is OOP?

- A programming paradigm based on objects and classes.
- Objects are instances of classes that contain attributes (data) and methods (functions).
- OOP allows for more modular, reusable, and maintainable code.

HyperionDev

Why Use OOP?

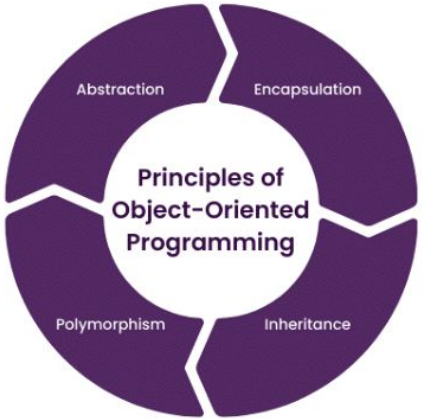
- Encourages modularity and reusability.
- Enhances code organization and maintenance.
- Helps in designing complex software systems.



HyperionDev

Key OOP Principles

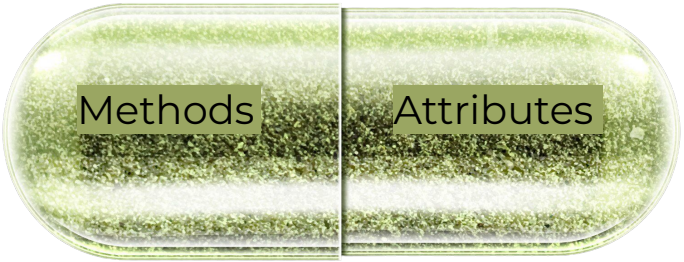
- Encapsulation
- Abstraction
- Inheritance
- Polymorphism



HyperionDev

Encapsulation

- Definition: Bundling data and methods inside a class while restricting direct access.
- Analogy: A capsule (medicine) hides its contents.
- Why? Protects data integrity, improves security, and maintains code structure.



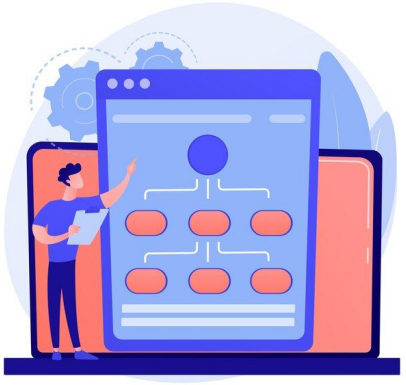
HyperionDev

Abstraction

- Definition: Hiding implementation details while showing only necessary functionality.
- Analogy: A car dashboard – you interact with controls without seeing internal mechanisms.
- Benefits: Reduces complexity, increases flexibility.

Inheritance

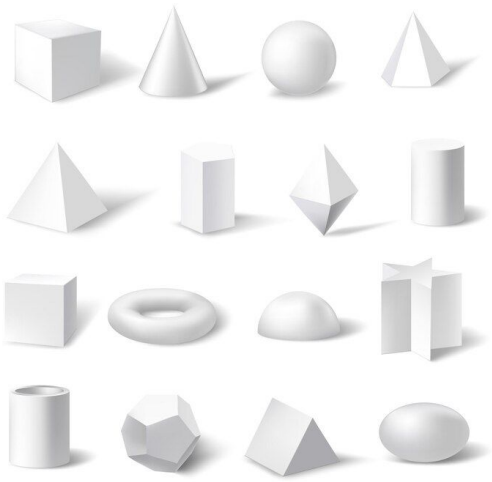
- Definition: Creating new classes from existing ones (parent-child relationship).
- Analogy: A child inherits traits from parents.
- Why? Reduces redundancy, promotes code reuse.



HyperionDev

Polymorphism

- Definition: One interface, multiple implementations.
- Analogy: A person behaves differently as a student, employee, and friend.
- Example: Method Overloading and Method Overriding.



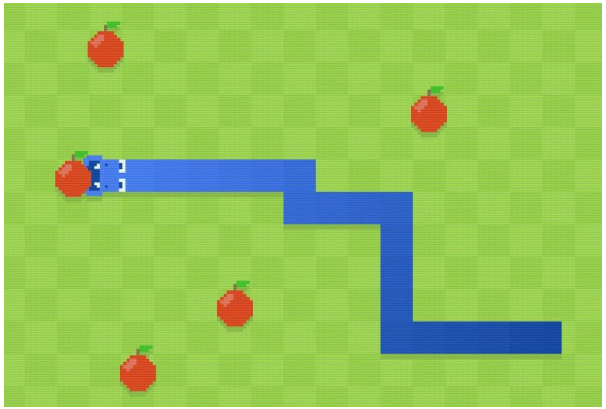
HyperionDev

Advantages & Disadvantages of OOP

- Advantages:
  - Code reusability
  - Scalability
  - Maintainability
- Disadvantages:
  - Higher complexity
  - Can have performance overhead
  - Can be overused leading to unnecessary abstraction

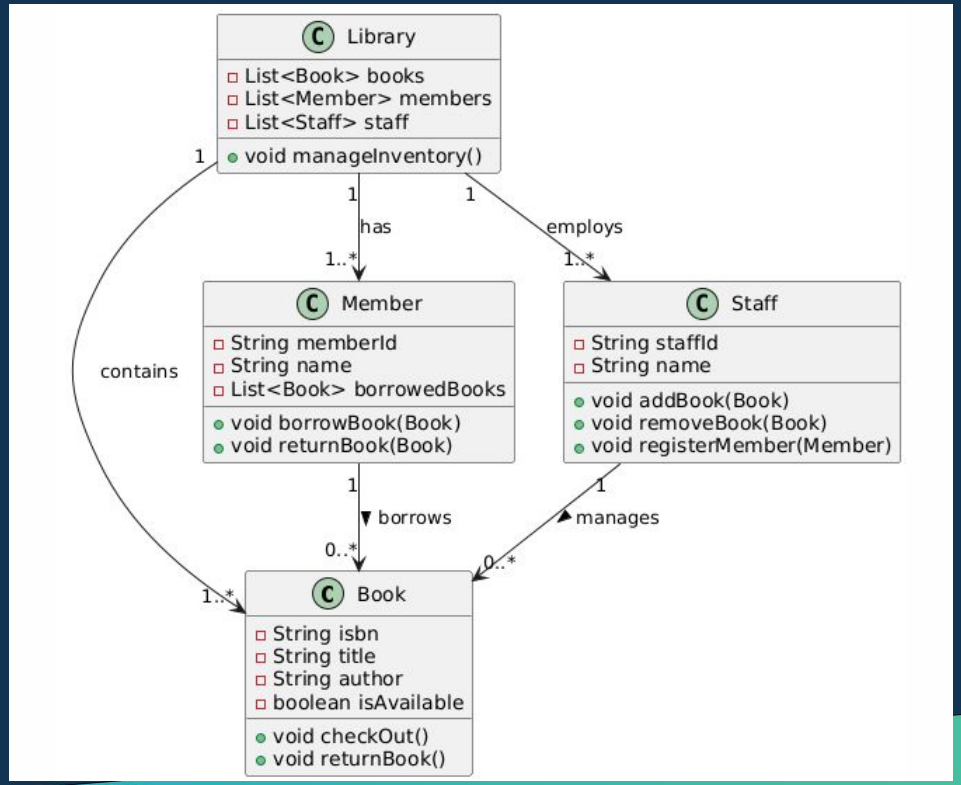
OOP in the Real World

- GUI Applications (Java Swing, Tkinter)
- Game Development (Unity, Unreal Engine)
- Web Applications (Django, Flask)



HyperionDev

Library Management System



HyperionDev

Recap

- OOP is centered around objects and classes.
- Encapsulation protects data.
- Abstraction hides details.
- Inheritance promotes code reuse.
- Polymorphism allows flexibility.

HyperionDev

Polls

Please have a look at the poll notification and select an option.

What is the main purpose of encapsulation in OOP?

A. To allow global access to class members
B. To hide the internal details of an object and expose only necessary parts
C. To ensure multiple classes share the same data
D. To increase the number of functions in a class

HyperionDev

Polls

Please have a look at the poll notification and select an option.

Which OOP principle allows a class to inherit properties from another class?

A. Encapsulation
B. Abstraction
C. Inheritance
D. Polymorphism

HyperionDev

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.


Thank you
for attending
HyperionDev