

Practice and Revision - Encapsulation and Inheritance

Chapter 7: Encapsulation and Inheritance · Topic 2

Key Concept

Encapsulation involves making class variables private and providing public methods (getters and setters) to access and modify them safely.

Inheritance allows a subclass (child class) to inherit properties and methods from a superclass (parent class), promoting code reuse.

The 'extends' keyword is used in Java to establish inheritance between a subclass and its superclass.

Understanding encapsulation and inheritance helps us write organised, reusable, and secure object-oriented programs.

Quick Revision Notes

- Encapsulation makes variables private.
- Public getters/setters access and modify them safely.
- Inheritance lets subclasses inherit from superclasses.
- This promotes code reuse.
- The 'extends' keyword establishes inheritance.
- Subclass = child class; superclass = parent class.
- These are key OOP principles.
- They help write organised, reusable, secure code.

Practice Questions

- 1 Write a class with private variables and public getters/setters.
- 2 Create a subclass that extends a superclass.
- 3 Discuss why encapsulation is important.
- 4 Explain how inheritance promotes code reuse.
- 5 Create your own example of a superclass and subclass.
- 6 Discuss the benefits of these OOP principles.