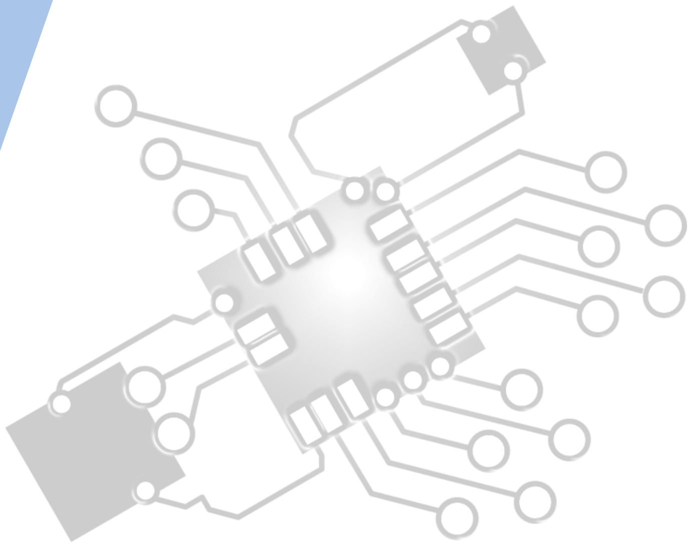
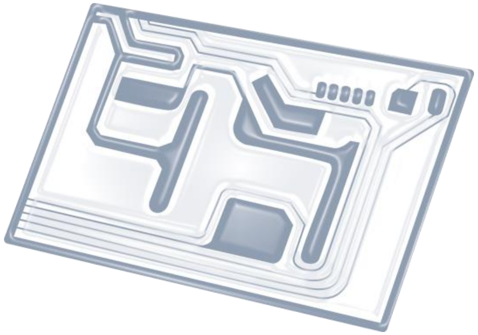
Features of OOP

IB Computer Science



Content developed by
Dartford Grammar School
Computer Science Department

HL Topics 1-7, D1-4



1: System design

2: Computer Organisation

3: Networks

4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP

Content developed by Dartford Grammar School Computer Science Department

HL & SL D.2 Overview

D.2 Features of OOP

D.2.1 Define the term encapsulation

D.2.2 Define the term inheritance

D.2.3 Define the term polymorphism

D.2.4 Explain the advantages of encapsulation

D.2.5 Explain the advantages of inheritance

D.2.6 Explain the advantages of polymorphism

D.2.7 Describe the advantages of libraries of objects

D.2.8 Describe the disadvantages of OOP

D.2.9 Discuss the use of programming teams

D.2.10 Explain the advantages of modularity in program development

1: System design

2: Computer Organisation

3: Networks

4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP

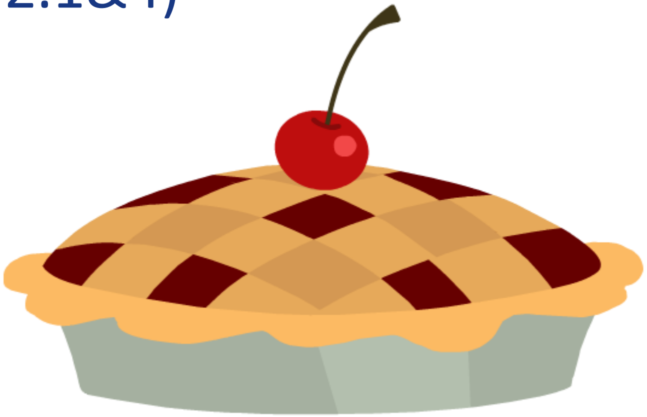
Content developed by Dartford Grammar School Computer Science Department

Topic D.2.5

Explain the **advantages** of **inheritance**

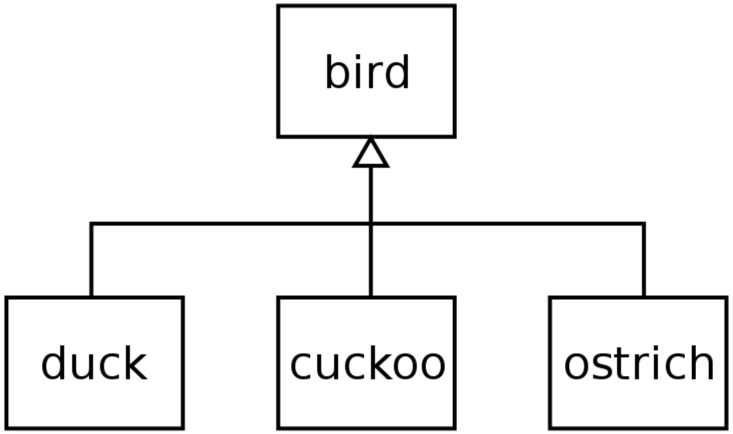
Four OOP fundamentals:

- **A**bstraction (*See* Topic 4.1.17-20)
- **P**olymorphism (*See* Topic D.2.3&6)
- **I**nheritance (*See* Topic D.2.2&5)
- **E**ncapsulation (*See* Topic D.2.1&4)



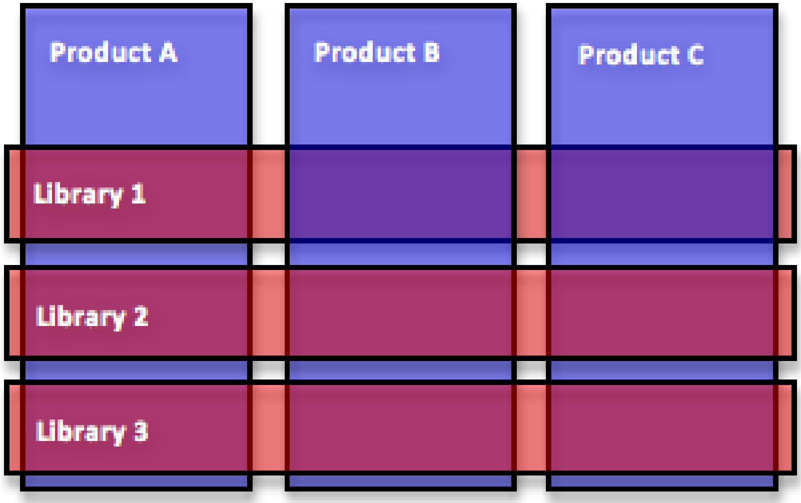
Definition: Inheritance

- Process whereby one object **inherits the properties** (states and behaviours) of another object (pairs called **super/sub** or **parent/child classes**)
- The Java keyword that implies inheritance is **`extends`**



Key benefit

Minimizing the amount of **duplicate code** in an application by **sharing common code** amongst several subclasses.

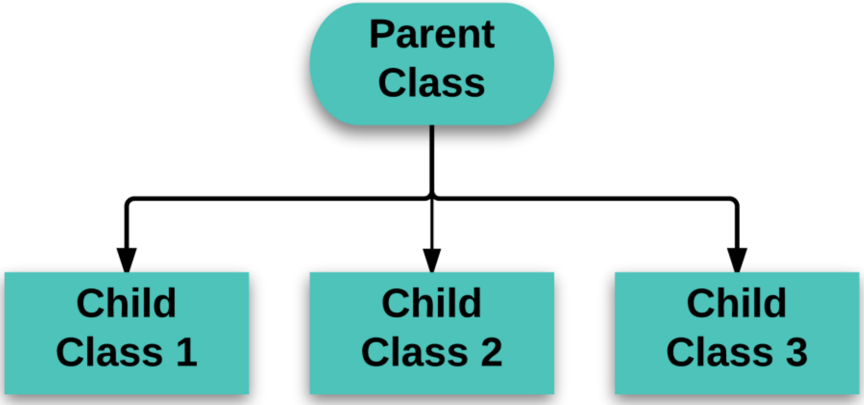


Other advantages of Inheritance

A. Minimize the amount of duplicate code in an application

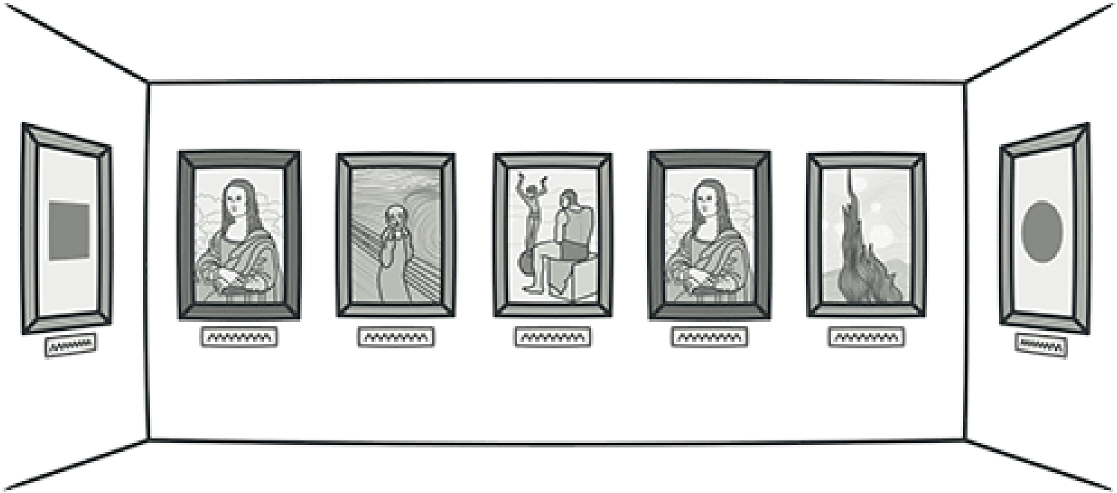
B. Better organization of code

C. Code more flexible change



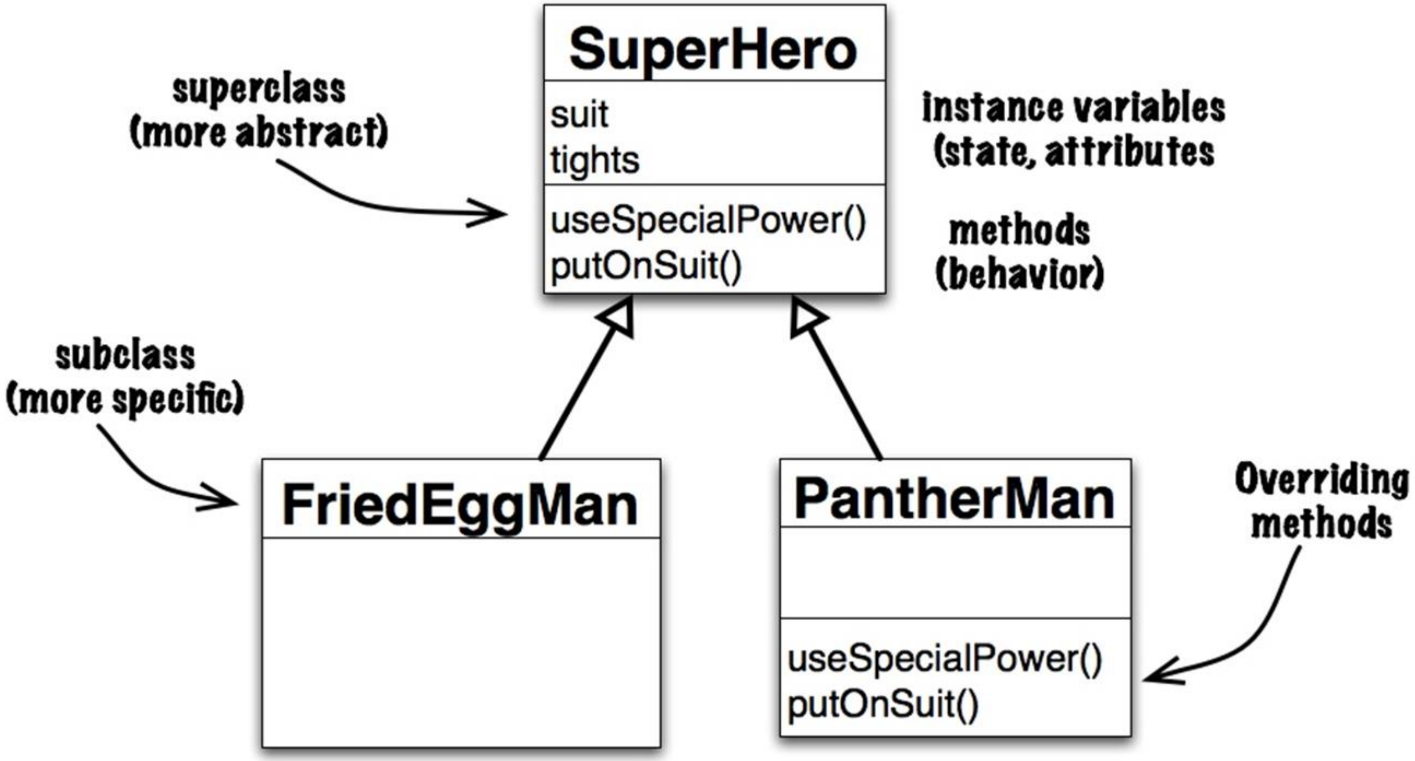
A. Minimize duplicate code

If duplicate code (variables and methods) exists in two related classes, they can be refactored into a hierarchy by moving that common code up to a common superclass.



B. Better organisation of code

Moving common code to a super class results in better organization of code (better abstraction).



C. Code more flexible to change

Inheritance can also make application code more flexible to change because classes that inherit from a common super class can be used interchangeably.

Exam note!

It is important to keep in mind that a **parent object** holds **common data** and **actions**, which **enhances reuse** and **reduces maintenance overheads**.