



# *Objects as a programming concept*

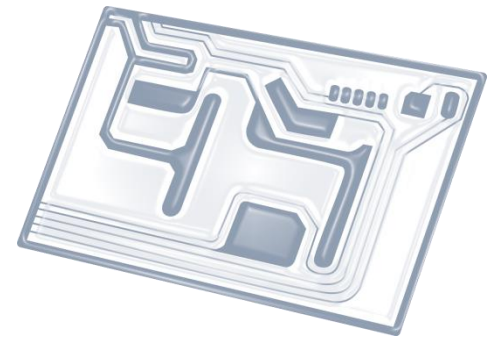
IB Computer Science



*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

# HL & SL D.1 Overview

## D.1 Objects as a programming concept

D.1.1 Outline the general nature of an object

D.1.2 Distinguish between an object (definition, template or class) and instantiation

D.1.3 Construct unified modelling language (UML) diagrams to represent object designs

D.1.4 Interpret UML diagrams

D.1.5 Describe the process of decomposition into several related objects

D.1.6 Describe the relationships between objects for a given problem

D.1.7 Outline the need to reduce dependencies between objects in a given problem

D.1.8 Construct related objects for a given problem

D.1.9 Explain the need for different data types to represent data items

D.1.10 Describe how data items can be passed to and from actions as parameters



1: System design

2: Computer Organisation



3: Networks

4: Computational thinking



5: Abstract data structures

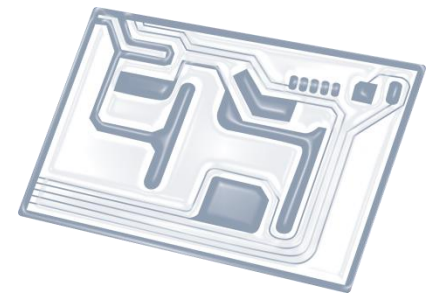
6: Resource management



7: Control

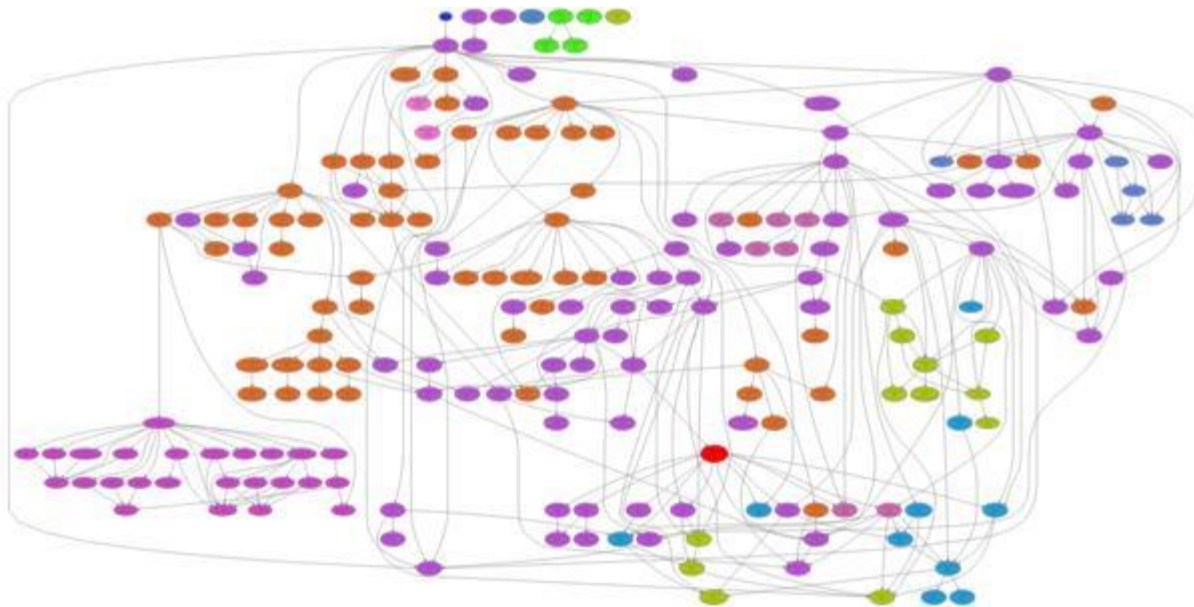
D: OOP





# Topic D.1.7

Outline the need to **reduce dependencies** between objects in a given problem



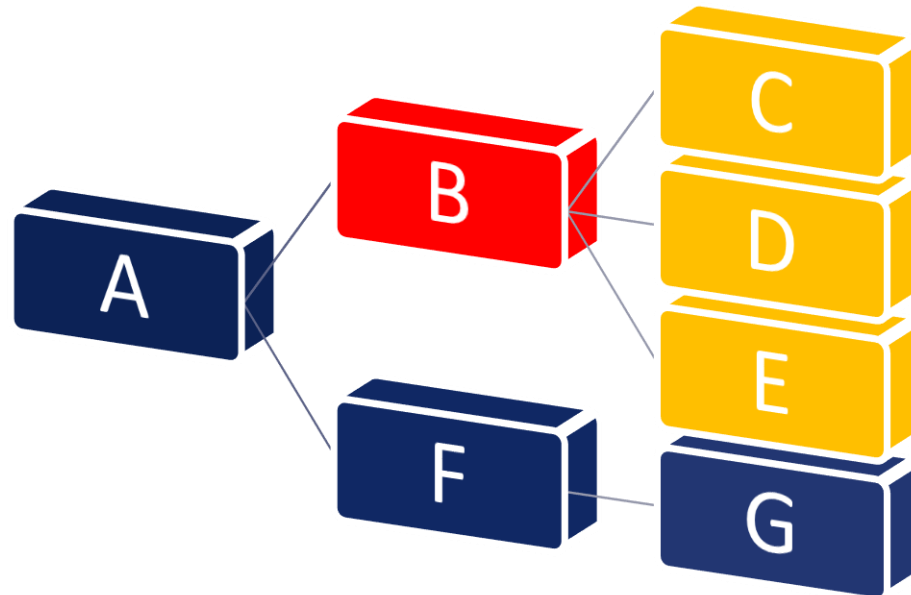
# Dependency? Why so bad?

- It **increases maintenance overheads**
- Maintenance overheads refer to the *changes that need to be made to the entire system if you make a change to a component.*

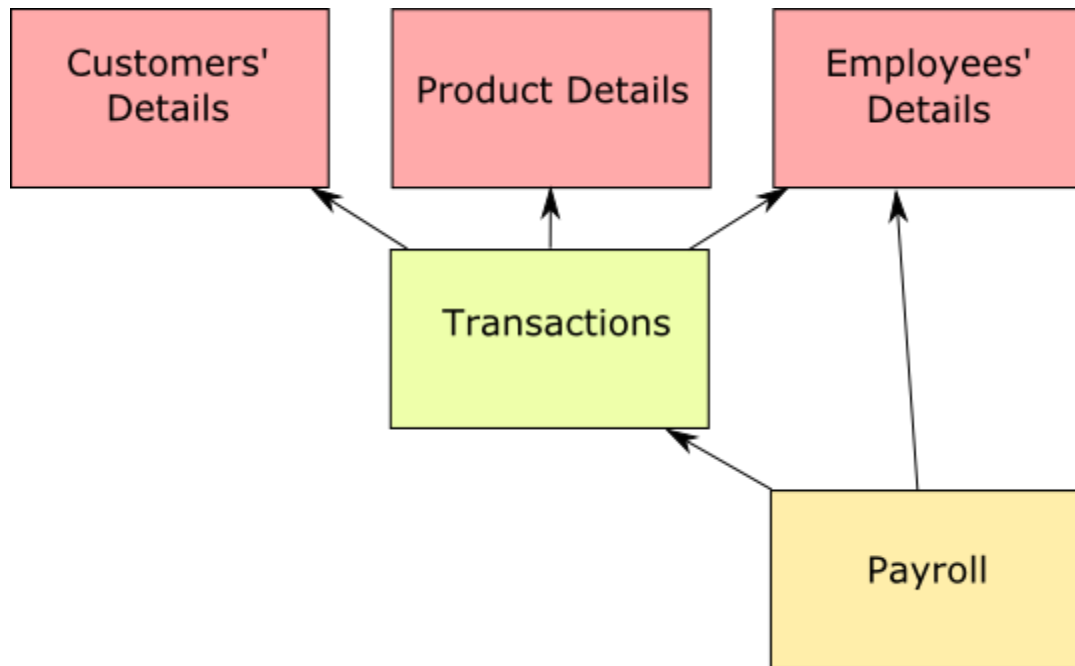


# Example

- For example if you change, B, it affects how C, D and E works, implying that you have to spend time fixing them to work with the “new” B.



# Imagine the changes needed!



- How much maintenance would it create if you change the **Customers' Details** class?
- How much maintenance would it create if you change the **Transactions** class?