



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



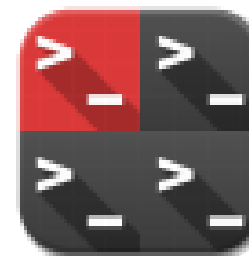
5: Abstract data structures



6: Resource management



7: Control



D: OOP

HL & SL D.3 Overview

D.3 Program development

D.3.1 Define the terms: class, identifier, primitive, instance variable, parameter variable, local variable

D.3.2 Define the terms: method, accessor, mutator, constructor, signature, return value

D.3.3 Define the terms: private, protected, public, extends, static

D.3.4 Describe the uses of the primitive data types and the reference class string

D.3.5 Construct code to implement assessment statements

D.3.6 Construct code examples related to selection statements

D.3.7 Construct code examples related to repetition statements

D.3.8 Construct code examples related to static arrays

D.3.9 Discuss the features of modern programming languages that enable internationalization

D.3.10 Discuss the ethical and moral obligations of programmers



1: System design

2: Computer Organisation



3: Networks

4: Computational thinking



5: Abstract data structures

6: Resource management

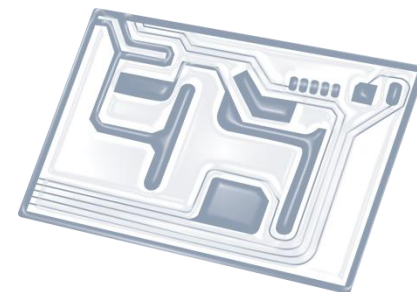


7: Control

D: OOP



Topic D.3.5




Construct code to implement **assessment statements** in D.3.1 to D.3.4



The only way to learn is to PRACTICE

- ✓ Class
- ✓ Identifier
- ✓ Primitive (data type)
- ✓ Instance variable
- ✓ Parameter variable
- ✓ Local variable
- ✓ Method
- ✓ Accessor
- ✓ Mutator
- ✓ Constructor
- ✓ Signature
- ✓ Return value
- ✓ Private modifier
- ✓ Protected modifier
- ✓ Public modifier
- ✓ Extends
- ✓ Static

A dark blue thought bubble with three smaller circles leading to it from the bottom right.

Make a set of classes to practice these concepts

```
public class People //The begin of the declaration of the class
{
    private int age; //Note the private keyword restrict access to variable only wit
    private string name;
    private int phone;

    new People() //The constructor method is used to initialise some values when an
    {
        age = 0;
        name = "";
        phone = 0;
    }

    public void setAge(int x) //The setter method is used to modify values of variab
    {
        age = x;
    }

    public int getAge() //The accessor method (part of the mutator methods) is used
    {
        return age;
    }

    public void setPhone(int x) //The next two methods are overloaded, the parameter
    {
        phone = x;
    }

    public void setPhone(string x)
    {
        phone = Transform(x);
    }

    private int Transform(string number) //Private keyword restricts use of method f
    {
        int numerical = Convert.ToInt(number); //local variable
        return numerical; //Return specifies what value the method returns to its ca
    }
}
```