

## Revision Guide for A-Level Computer Science Unit 1.2.4 (SLR 07)

### 1.2.4 Type of Programming Language

Candidates need to understand that there are a variety of types of programming paradigms such as procedural, OOP, low-level, and that each has its strengths and weaknesses in specific scenarios, topics or areas.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need to have knowledge and experience of using a procedural programming language for example Python, VB.NET etc.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need to be experienced in using procedural programming features such as (but not limited to) variables, constants, selection, iteration, sequence, subroutines, string handling, file handling, Boolean and arithmetic operators.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need to be able to read, trace, amend and write procedural program code.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need to have an understanding of the purpose and need for assembly language. They need to be familiar with the instructions given in Appendix 5d. They should be able to read, write, trace and amend programs written in the Little Man Computer language.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need an understanding of addressing, which should be integrated with assembly language.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates should have experience of using immediate, direct, indirect and indexed addressing in the writing, reading and tracing of programs written in assembly language.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need to understand object-oriented code (as specified in the pseudocode guide). They need to have an understanding of classes, objects, attributes and methods. They need to understand the difference between private and public attributes and methods.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need to understand encapsulation and the use of get and set methods to access private attributes.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need to understand the purpose and principles of inheritance.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need to have an understanding of polymorphism and how it can be used within a program.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆
Candidates need to be able to read, trace, amend and write code that makes use of these object-oriented techniques.	Before Revision☆☆☆☆☆ After Revision☆☆☆☆☆

Slideshow (Shared area)	PG Online Textbook	Teach-ICT	CraignDave (Youtube)
Slides 1 to 92  <b>Read</b> <input type="checkbox"/> <b>Starter Activities</b> <input type="checkbox"/> <b>Tasks</b> <input type="checkbox"/>	Section 3: Chapters 13 & 14 Pages 64 to 73  <b>Read</b> <input type="checkbox"/> <b>Questions</b> <input type="checkbox"/> <b>Exercises P68</b> <input type="checkbox"/> <b>Exercises P73</b> <input type="checkbox"/>	1.2.4 Types of Programming language – Programming Paradigms <b>Theory</b> <input type="checkbox"/> 1.2.4 Types of Programming language – Procedural Language Techniques <b>Theory</b> <input type="checkbox"/> 1.2.4 Assembly Language – Little Man Computer <b>Theory</b> <input type="checkbox"/> <b>Lesson Tasks</b> <input type="checkbox"/> 1.2.4 Assembly Language – Memory Modes <b>Theory</b> <input type="checkbox"/> 1.2.5 – Object Orientated Programming - OOP <b>Theory</b> <input type="checkbox"/> <b>Lesson Tasks</b> <input type="checkbox"/>	OCR A'Level Programming paradigms <b>Watched</b> <input type="checkbox"/> <b>Cornell Notes</b> <input type="checkbox"/>  OCR A'Level Procedural language <b>Watched</b> <input type="checkbox"/> <b>Cornell Notes</b> <input type="checkbox"/>  OCR A'Level Assembly language and LMC language <b>Watched</b> <input type="checkbox"/> <b>Cornell Notes</b> <input type="checkbox"/>  OCR A'Level Addressing memory <b>Watched</b> <input type="checkbox"/> <b>Cornell Notes</b> <input type="checkbox"/>  OCR A'Level OO languages part 1-part 4 <b>Watched</b> <input type="checkbox"/> <b>Cornell Notes</b> <input type="checkbox"/>