- Programming language:
    - A formal language for describing computation
    - A “user” interface to a computer
    - Syntax and semantics
    - Compiler, translator, and interpreter
    - A tool to support a programming paradigm

- Assembly code = low level programming language. High = java

- The different types of programming language styles are:
    - **Imperative style**: deal with algorithms + data and both are equal importance. Good for **decomposition** (tackle problem into pieces) eg: pascal and c
    - **Functional style**: for functions. Good for **reasoning** eg: liso and sml
    - **Logical programming styles**: based on facts and rules. Good **for searching** eg: prolog
    - **Objective oriented programming**: objects have attribute and activities and interact with others. Good for **modelling** eg: C++, Java

- Good programming language: (need, user, application)
    - **Ease of design and coding:** so full control and easy to understand
    - **Debugging:** find all mistakes in trial room
    - **Maintenance:** deploy a program
    - **Reusability:** codes (are changeable, recycle, flexible and add more functionalities)

- Criteria for designing a good programming language:
    - **Readability**: Make sure the programming language is understood, comprehended easily and accurately
    - **Write-ability:** so when you write code it needs to be nicer and clean so understandable
    - **Reliability:** assure a coded program will not behave in an unexpected or disastrous way. So generate same function every time
    - **Uniformity**: features should look similar & behave similar
    - **Maintainability:** make sure errors can be found and corrected and new features for the language can be added. So error checking

- A programming language has components:
    - Interpreter: basic form of program that will convert our programs to some output. Eg: source code will be converted to output by translator.
    - Compilation: source file goes through compiler to generate assembly file.

- How will a high language programming language will be converted to machine code:
    - Step 1: Source code may go through the pre-processor to convert the source code to pre-processed code.
    - Step 2: The pre-processed code will get compiled by compiler to assembly code.
    - Step 3: Assembly code will be go through assembler. Therefore, object code file is created.

- Step 4: The object file is linked and give filename.obj. Environment for code is given
- Step 5: Linker: all the other object files, and library files are linked.
- Step 6: exe files created. Stored in secondary storage ie harddisc
- Step 7: The most efficient way to run an executable is the exe will be loaded by loader to primary memory to be able to run