



Section 16.2 - Translation Software

Layer 4: Operating System

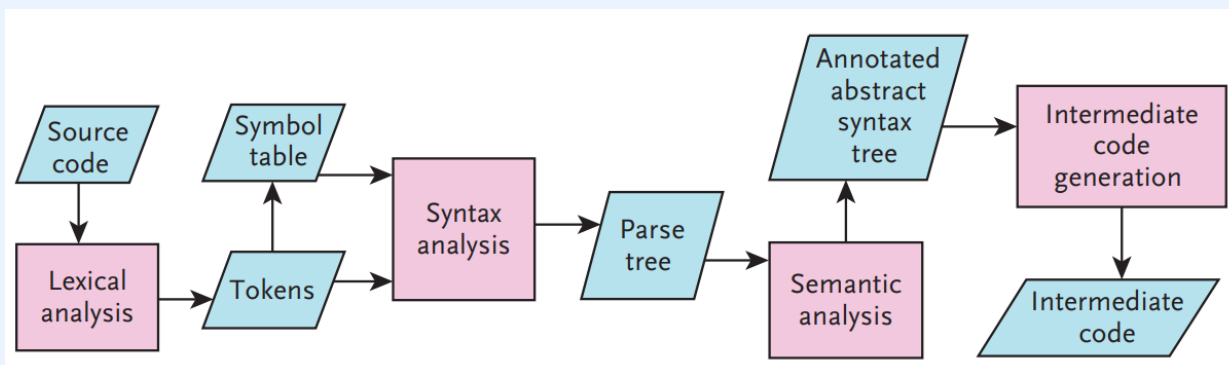
Syllabus Content Section 16: System Software

 S16.2.1 Show understanding of how an interpreter can execute programs without producing a translated version 

compiler	Interpreter
Translates a high-level language program to machine code.	Translates and executes a high-level language program, line-by-line.
Creates a .exe file which can be easily distributed.	No .exe file created.
Once compiled, .exe file does not need to be compiled again, resulting in faster execution.	Execution very slow – translated each time program run.
Reports all errors at the end of compilation: difficult to locate errors.: development process long.	Debugging easier/faster, since it stops translating when it reaches an error. This allows real time error correction.
Only be produced when all errors are fixed.	Can run program any time, even before code finished.
Used when development is completed.	Used during development.

 S16.2.2 Show understanding of the various stages in the compilation of a program 

- Including lexical analysis, syntax analysis, code generation and optimisation



S16.2.3 Show understanding of how the grammar of a language can be expressed using syntax diagrams or Backus-Naur Form (BNF) notation

< >	means can be described as
::=	means the thing on the left of this is made up by the things from the right
	means or
*	means the thing may happen or not happen
{ }	means anything inside must be grouped together
[]	means anything inside is optional
	means a space (empty space)
_	also means a space

S16.2.4 Show understanding of how Reverse Polish Notation (RPN) can be used to carry out the evaluation of expressions

Example

A username must:

Start with one upper case letter from A to E

Then it should have two numbers from 0 to 9

Then it should have three lower case letters from f to k

Then it should have a symbol from ? ! * & ^ £ \$

```
<Username> ::= <UpperCase><Number><Number><LowerCase><Symbol>
```

But you have to say what is <UpperCase>....

```
<UpperCase> ::= A|B|C|D|E
```

```
<Number> ::= 0|1|2|3|4|5|6|7|8|9
```

```
<LowerCase> ::= f|g|h|i|j|k
```

```
<Symbol> ::= ?|!|*|&|^|£|$
```