**OYEBADE OLUWARTOIMI**

**17/SCI01/074**

**CSC 310 ASSIGNMENT(translators)**

1. **TRANSLATORS**

A translator is a programming language processor that converts a computer program from one language to another. It takes a program written in source code and converts it into machine code. It discovers and identifies the error during translation.

It translates high-level language program into a machine language program that the central processing unit (CPU) can understand. It also detects errors in the program.

1. **(a) ASSEMBLER**

An assembler is is a translator used to translate assembly language to machine language. It is like a compiler for the assembly language but interactive like an interpreter. Assembly language is difficult to understand as it is a low-level programming language. An assembler translates a low-level language, an assembly language to an even lower-level language, which is the machine code. The machine code can be directly understood by the CPU.

 **(b) COMPILER**

A compiler is a translator used to convert high-level programming language to low-level programming language. It converts the whole program in one session and reports errors detected after the conversion. Compiler takes time to do its work as it translates high-level code to lower-level code all at once and then saves it to memory.

A compiler is processor-dependent and platform-dependent. But it has been addressed by a special compiler, a cross-compiler and a source-to-source compiler. Before choosing a compiler, user has to identify first the Instruction Set Architecture (ISA), the operating system (OS) and the programming language that will be used to ensure that it will be compatible.

 **(c) INTERPRETER**

Just like a compiler, is a translator used to convert high-level programming language to low-level programming language. It converts the program one at a time and reports errors detected at once, while doing the conversion. With this, it is easier to detect errors than in a compiler. An interpreter is faster than a compiler as it immediately executes the code upon reading the code.

It is often used as a debugging tool for software development as it can execute a single line of code at a time. An interpreter is also more portable than a compiler as it is not processor-dependent, you can work between hardware architectures.

1. **THE NEED FOR HIGH LEVEL PROGRAMMING LANGUAGE**

The main advantage of high-level languages over low-level languages is that they are easier to read, write, and maintain. Ultimately, programs written in a high-level language must be translated into machine language by a compiler or interpreter.

In computer science, a high-level programming language is a programming language with strong abstraction from the details of the computer. In contrast to low-level programming languages, it may use natural language elements, be easier to use, or may automate (or even hide entirely) significant areas of computing systems (e.g. memory management), making the process of developing a program simpler and more understandable than when using a lower-level language. The amount of abstraction provided defines how "high-level" a programming language is.