**GABRIEL-OHANU VICTOR**

**17/sci01/035**

**Csc310**

**Q1**

A translator is a programming language processor that converts a [computer program from one language](https://teachcomputerscience.com/programming-languages/%22%20%5Ct%20%22_blank) to another.  It takes a program written in source code and converts it into machine code.  It discovers and identifies the error during translation.

**Q2**

|  |  |  |
| --- | --- | --- |
| **INTERPRETOR** | **COMPILER** | **ASSEMBLER** |
| Translates high level language into machine code  | Translates high level language into machine code  | Translates assembly language into machine code  |
| Translates a source one line at a time  | Translates all code at one time  | Uses the processors instruction set to convert |
| Needed everytime you run the program  | Only needed to create an executable file | Runs quickly as conversation between two low level languages is reliant on the processors instruction set |
| Returns a list of errors found and on which lines  | Will only inform you of the first error found |  |
| Runs more slowly as it is being translated everytime the code is run  | Onced compiled runs quickly but compiling can take a long time |  |

**Q3**

1. High-level languages are programmer-friendly. They are easy to write, debug and maintain.
2. It provides a higher level of abstraction from machine languages.
3. It is a machine-independent language.
4. Easy to learn.
5. Less error-prone, easy to find and debug errors.
6. High-level programming results in better programming productivity.