WEEK 2

Date: 20/01/2020

TOPIC: Programming Language.

LEARNING OBJECTIVES

At the end of the lesson students should be able to:

- Identify a computer program;
- Define a computer programming language;
- Give examples of computer programming languages

INTRODUCTION

A computer needs written instructions to perform specific tasks. These written instructions are called programs. Technically, a program is defined as a sequence of instructions written to perform a specified task with a computer. It is an organised list of instructions that, when executed causes the computer to behave in a predetermined manner.

DEFINITION OF PROGRAMMING LANGUAGE

A programming language is an artificial language that is designed to communicate instructions to a machine, particularly a computer. A programming language can be used to create programs which control the behaviour of a machine particularly a computer.

A programming language can be described from the point of view of the form, called the *syntax* or from the point of view of the meaning called the *semantics*.

There are several programming languages that have been created. Some of these programming languages may have different dialects.

Examples of programming languages are:

1. BASIC, 2. FORTRAN, 3. COBOL, 4. C, 5. Pascal, 6. Algol,
7. Logo, 8. Java, 9. Prolog, 10. SNOBOL, 11. Visual Basic

Program Errors

Certain errors occur in the course of programming. They include:

Syntax error: In any language such as English, there are sets of rules which govern the structure of the statements. When these rules are not followed, then you can have syntax error. These errors can occur when wrong punctuations are used.

Logical error: Each record is made up of a batch of records called *logical records*. When there is a misuse of this arrangement, logical error can occur.

Another type of error is the **run-time error.**

SUB-TOPIC 1: TYPES OF PROGRAMMING LANGUAGES

There are many types of computer languages. The levels, purpose and structures can differentiate types of languages.

There are basically three types of computer programming languages**:**

- **Machine language or machine code:** is the only language understood by the computer. Each statement in a machine language is a sequence of bits (0 or 1). It is a low-level programming

language. While a machine language easily understood by computers, it is very difficult to understand by people. This is where a high-level programming language comes in.

Programs written in high-level languages are easily understood by people and so when programs written in high-level languages are compiled and/or interpreted in machine language, computers can easily execute them.

- **Assembly language: is a** representation of machine language in the sense that each instruction of assembly language translates an instruction in a machine code. The advantage an assembly language has over a machine language is that, its instructions are readable. Although an assembly language is more readable than a machine language, it is still a low-level language. Another disadvantage of assembly language is that it is not portable in the sense that it is specific to a particular hardware. For example, the assembly language on Mac will not work on Personal Computers (PC).
- **High-level language:** are languages that are used by most programmers. Examples of high-level languages are BASIC, FORTRAN and COBOL. A great advantage of high-level languages is that they are very readable because they use English-like words. Another advantage of high-level languages is that they it is less tedious to use and it is usually portable as it can work on different platforms. High-level languages however are usually less powerful and less efficient; they need to be compiled and/or interpreted into machine code before they are executed. Examples of high-level languages are: C, C++, Pascal, COBOL, FORTRAN, LISP, RPG