

**PENANG SANGAM HIGH SCHOOL
P.O.BOX 44, RAKIRAKI**

LESSON NOTES 14

Year/Level: 11 Subjects: Computer Studies

| | |
|----------------------------------|--|
| Strand: | CE 2Application package |
| Sub-strand: | Programming |
| Content Learning Outcome: | ❖ Step 5 – Program Documentation ❖ Step 6 – Program Maintenance |

Lesson Notes

STEP 5 - PROGRAM DOCUMENTATION

- Documentation consists of written descriptions and procedures about a program and how to use it.
- It not something done just at the end of the programming process.
- Documentation is important for all people who may be involved with the program in the future.
- These people include the following:
 - Users** – those who need to know how to use the software in organization
 - Operators** - Documentation must be provided for computer operators (in case of special purpose programs). When the operators encounter error messages while using these programs, they need to be able to refer to the appropriate documentation to encounter those error messages.
 - Programmers** - programmers wishing to update or modify it i.e. perform maintenance, may find themselves stuck without adequate documentation.

STEP 6 - PROGRAM MAINTENANCE

- The purpose of program maintenance is to ensure that the current programs are operating error-free, efficiently and effectively.
- Activities in this area fall in two categories:
 1. **Operations** - Concerns locating and correcting operational errors, making programs easier to use and standardizing software using structured programming techniques.
 2. **Changing Needs** - All organizations change over time and their programs must change with them. Programs need to be adjusted for a variety of reasons including new tax laws, new information needs and new information policies etc.

Five Generations of Programming Languages

There are five generations of programming languages. These are:

1. Machine Languages: The First Generation

- a byte is made up of bits, consisting of 0's and 1's.
- These 0's and 1's may correspond to electricity's being on or off in the computer.
- Examples of these coding schemes, as we saw, are ASCII and EBCDIC.
- Data represented in 0's and 1's is said to be written in machine language. To see how hard this is to understand, imagine if you had to code this:

111100100111001111010010000100000111000000101011

2. Assembly Languages: The Second Generation

- Assembly languages have a clear advantage over the 0's and 1's of machine language because

they use abbreviations or mnemonics.

- These are easier for human beings to remember.
- The machine language code we gave above could be expressed in assembly language as
ADD 210 (8, 13), 028 (4, 7)
- This is still pretty obscure, of course, and so assembly language is also considered low-level.

3. High-Level Procedural Languages: The Third Generation

- People are able to understand languages that are more like their own (e.g., English) than machine languages or assembly languages.
- These more English-like programming languages are called "high-level" languages.
- Procedural languages are programming languages with names like BASIC, Pascal, C, COBOL, and FORTRAN.

Compilers and Interpreters

For a procedural language to work on a computer, it must be translated into machine language so that the computer understands it. Depending on the language, this translation is performed by either a compiler or an interpreter.

A **compiler** converts the programmer's procedural language program, called the source code, into a machine language code, called the object code. This object code can then be saved and run later.

An **interpreter** converts the procedural language one statement at a time into machine code just before it is to be executed. No object code is saved. An example of a procedural language using an interpreter is the standard version of BASIC.

4. Problem-Oriented Languages: The Fourth Generation

Problem-oriented languages, also known as very high level languages, require little special training on the part of the user. Unlike general-purpose- languages, problem-oriented languages are designed to solve specific problems. This group also includes query languages and application generators:

- *Query languages:* Query languages enable nonprogrammers to use certain easily understood commands to search and generate reports from a database. An example is the commands used on an airline reservations system by clerks needing flight information.
- *Application generators:* An application generator contains a number of modules-logically related program statements - that have been pre-programmed to accomplish various tasks. An example would be a module that calculates over-time pay.

5. Natural Languages: The Fifth Generation

Natural languages are still being developed. They are designed to give people a more human (—naturall) connection with computers. The languages are human languages: English, French, Japanese, or whatever.

Questions

- 1) Name the two activities carried out in Program Maintenance?
- 2) Which programming language does BASIC, Pascal, C, COBOL, and FORTRAN fall.