Programming

What exactly is programming? Many people think of it as simply typing words into a computer. That may be part of it, but that is certainly not all of it. Programming is actually a *problem-solving procedure.*

A program is a list of instructions for the computer to follow to process data. Programming, also known as software development, is a six-step procedure for creating that list of instructions. Only one of those steps consists of typing (keying) statements into a computer.

- **Program specification**: The program's objectives, outputs, inputs, and processing requirements are determined. It is also called program definition or program analysis.
- **Program design**: A solution is created using programming techniques such as top-down program design, pseudocode, flowcharts, and logic structures.
- **Program code**: The program is written or coded using a programming language.
- **Program test**: The program is tested or debugged by looking for syntax and logic errors.
- **Program documentation**: Documentation is an ongoing process throughout the programming process. This phase focuses on formalizing the written description and processes used in the program.
- **Program maintenance**: Completed programs are periodically reviewed to evaluate their accuracy, efficiency, standardization, and ease of use. Changes are made to the program's code as needed.

Computer languages

Unfortunately for us, computers can't understand spoken English or any other natural language. Computers typically cannot tolerate ambiguity or mistakes. Statements (instructions) must be unambiguous and without any errors. The only language they can understand directly is machine code, which consists of 1's and 0's (binary code). Machine code is too difficult to write. For this reason, we use symbolic languages to communicate instructions to the computer. For example, assembly languages use abbreviations such as ADD, SUB, MPY to represent instructions. The program is then translated into machine code by a piece of software called an assembler. Machine code and assembly languages are called low-level languages because they are closer to the hardware. They are quite complex and restricted to particular machines. To make the programs easier to write, and to overcome the problem of intercommunication between different types of computer, software developers designed high-level languages, which are closer to the English language. Programs written in high-level languages must be translated into machine code by a compiler or an interpreter. A compiler translates the source code into object code - that is, it converts the entire program into machine code in one go. On the other hand, an interpreter translates the source code line by line as the program is running.

Read the text again and answer these questions.

1. What are the 6 steps in programming?
2. Which one do you prefer? Why? Why not?
3. Do computers understand human languages? Why?/ Why not?
4. What is the function of an assembler?
5. Why did software developers design high-level languages?
6. What is the difference between a compiler and an interpreter?

PROGRAMMING

Steps in programming

Listen to a software developer, talking to a group of students on a training course about how a program is written

Put these steps into the correct order.

Write instructions in a programming language
Prepare documentation
Understand the problem and plan a solution
Make a flowchart of the program
Compile the program (to turn it into machine code)
Test and debug the program

Match the words (1-7) with the definitions (a-g).

1. Algorithm
2. flowchart
3. source code
4. compiler
5. machine code
6. debugging
7. coding

a) Program instructions written in a particular computer language
b) The techniques of detecting and correcting errors (or bugs) which may occur in programs
c) A diagram representing the successive logical steps of the program
d) A special program which converts the source program into machine code - the only language understood by the processor
e) The basic instructions understood by computers; it consists of l's and 0's (binary code)
f) a set of steps that are followed in order to solve a mathematical problem or to complete a computer process
g) to translate (a program) into language that can be communicated to the computer The process of writing instructlons.

Complete the following with one of the words in bold.

program	programmers	programming	programmable

1. ……..is the process of writing a program using a computer language.
2. A computer………… is a set of instructions that tells the computer how to do a specific task.
3. Most computer…………. make a plan of the program before they write it
4. A ……………..keyboard allows the user to configure the layout and meaning of the keys.

compile	compiler	compilation

5. Programs written in a high-level language require ….. that is, translation into machine code, the language understood by the processor.
6. A source program is converted into machine code by software called a …….
7. Programmers usually ………their programs to generate an object program and diagnose possible errors.

bug	debug	debugger	debugging

8. Any error or malfunction of a computer program is known as a ……
9. A ……..is a program used to test and other programs.
10. The process of going through the code to identify the cause of errors and fixing them is…………..