Programming

Introduction

- The field of computer programming begins with the definition of a computer: “a programmable usually electronic device that can store, retrieve, and process data”.
- In plain language, a computer is a device that can perform mathematic and logical operations much faster than any human could (and accurately).
- A useful analogy can be made between humans and computers, as both of us have several input and output mechanisms, in the case of humans: eyes, ears, mouths, etc. On the other hand, computers have keyboards, mice, speakers, etc.

Basics

- All computer programs take place within the intangible elements of a computer; therefore, there is a need for a programming language to provide the necessary framework.
- A programming language is: "a computer's language of instruction and information". The information expressed in such a language is what we call "data".

Definitions

- **Booleans:** Data that can either be true or false, also represented by 0 or 1.
- **Char:** This is the name given to a character, usually a single letter, since although a number can be stored by this type of data it cannot be later modified by arithmetical operations applied to it.
- **Floats:** They are essentially numbers, but they can have decimals.
- **Integers:** They are the opposite of the previous type, they are numbers, but they cannot have decimals.

Definitions

- **Strings:** This data is basically a combination of several char, it can therefore store text of any type.
- Even though the different types of data have been defined this is not enough to build a first program. A program has two important elements, the variables or data types and the flow control.
- There are essentially three instructions that contribute to flow control in every single programming language:

Flow Control

- **If:** by far the most common way of controlling the actions of a program, this statement takes a decision based upon the comparison of a condition between one or more variables.
- **For:** a very common way of generating iterations, it generates a cycle that remains in effect until a condition becomes false.
- **While:** very similar to the option before, it is preferred when the code within the loop must be executed at least once.

Programming Definitions

- **Computer Programming:** Process of designing, writing, testing, debugging, and maintaining the source code of computer programs.
- **Debugging:** Methodical process of finding and reducing the number of bugs, or defects, in a computer program or a piece of electronic hardware, thus making it behave as expected.
- **Source Code:** Text written in a computer programming language, specially designed to specify the actions to be performed by a computer.
- **Computer Program:** Sequence of instructions written to perform a specified task with a computer.

Programming Definitions

- **Programming Language:** Artificial language designed to communicate instructions to a machine.
- **Algorithms:** A step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps.
- **Compiler:** Computer Program (or set of programs) that transforms source code written in a programming language into another computer language. The most common reason for wanting to transform source code is to create an executable program.

Programming Definitions

- **Readability:** Ease with which a human reader can comprehend the purpose, control flow, and operation of source code.
- **Operating system:** Software consisting of programs and data, that runs on computers, manages computer hardware resources, and provides common services for execution of various application software.
- **Software bug:** Term used to describe an error, flaw, mistake, failure, or fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unintended ways.

Programs Requirements

Whatever the approach to software development may be, the final program must satisfy some fundamental properties. The following properties are among the most relevant: efficiency, reliability, robustness, portability, usability, maintainability, and programming style.

Programs Requirements

- **How often the results of a program are correct:** This depends on conceptual correctness of algorithms, and minimization of programming mistakes, such as mistakes in resource management (e.g., logic errors such as division by zero or off-by-one errors).
- **How well a program anticipates problems not due to programmer error:** This includes situations such as incorrect, inappropriate or corrupt data, unavailability of needed resources such as memory, operating system services and network connections, and user error.

Programs Requirements

- **The ease with which a person can use the program for its intended purpose, or in some cases even unanticipated purposes:** This involves a wide range of textual, graphical and sometimes hardware elements that improve the clarity, intuitiveness, cohesiveness and completeness of a program's user interface.
- **The range of computer hardware and operating system platforms on which the source code of a program can be compiled/interpreted and run:** This depends on differences in the programming facilities provided by the different platforms, including hardware and operating system resources, expected behavior of the hardware and operating system, and availability of platform specific compilers (and sometimes libraries) for the language of the source code.

Programs Requirements

- The ease with which a program can be modified by its present or future developers in order to make improvements or customizations, fix bugs and security holes, or adapt it to new environments: Good practices during initial development make the difference in this regard. This quality may not be directly apparent to the end user but it can significantly affect the fate of a program over the long term.
- The amount of system resources a program consumes: This also includes correct disposal of some resources, such as cleaning up temporary files and lack of memory leaks.