Mr. Giansante

C++ Programming

Introduction

August 2018

Introduction to Programming

> The computer is no better than its program.
>
> Men, Machines and Modern Times (1966)
> Elting E. Morison

> I think that's a shame. There's a lot of reasoning skills that you learn by knowing how to program. But more than that, you get to see what you can do when you're the master. You have an idea, you type in a program, and it does what you want it to. Nowadays, we're taught to go and buy such and such a program to use it their way, the way they designed it. ...You're a slave to the way someone else does it for you.
>
> Steve Wozniak
> Co-founder of Apple Computer
> Palais des Congres in Montreal, Canada
> Thursday, February 15, 2007
> Commenting on the fact that school children aren't taught computer programming language anymore.

Introduction

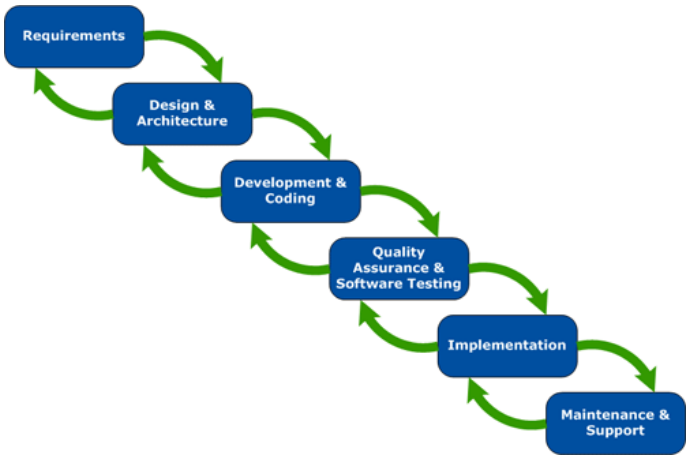
Although computers appear to be amazingly intelligent machines, they cannot yet think on their own. Computers still rely on human beings to give them detailed instructions on how to carry out even the most basic tasks. These instructions are called programs and are written using computer programming languages.

Computer programming (often shortened to programming or coding) is the process of writing, testing, debugging/troubleshooting, and maintaining the source code of computer programs. The code may be a modification of an existing source or something completely new. The purpose of programming is to create a program that exhibits a certain desired behaviour. The process of writing source code often requires expertise in many different subjects, including knowledge of the application domain, specialized algorithms and formal logic.

There is an ongoing debate on the extent to which the writing of programs is an art, a craft or an engineering discipline. Good programming is generally considered to be the measured application of all three, with the goal of producing an efficient and evolvable software solution.

Within software engineering, programming (the implementation) is regarded as one phase in a software development lifecycle.

Software Development Lifecycle



Quality Requirements

Whatever the approach to software development may be, the final program must satisfy some fundamental properties. The following five properties are among the most relevant:

Efficiency / Performance: the amount of system resources a program consumes (processor time, memory space, slow devices, network bandwidth and to some extent even user interaction), the less the better.

Reliability: how often the results of a program are correct. This depends on various methods of prevention of errors.

Robustness: how well a program anticipates situations of incompatibilities that result in run time errors and program halts.

Usability: the clarity and intuitiveness of a programs output can make or break its success. This involves a wide range of textual and graphical elements that makes a program easy and comfortable to use.

Portability: 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 mainly on the range of platform specific compilers for the language of the source code rather than anything having to do with the program directly.

Source: wikipedia.org

Proper Programming Style

Programs that are not easy to understand are bad programs even though they may give correct answers. Frequently, programs must be modified at a later date. If the person who is to modify the program cannot understand it, then they must start over and waste all the original programmer's effort. There are a number of ways to make computer programs more understandable.

Why Have Code Conventions?

Code conventions are important to programmers for a number of reasons:

80% of the lifetime cost of a piece of software goes to maintenance.

Hardly any software is maintained for its whole life by the original author.

Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly.

If you ship your source code as a product, you need to make sure it is as well packaged and clean as any other product you create.

Coding Conventions for this Class

Choose names for variables that describe the value they will be storing. For example: if you wish to have a variable store a person's total income, call the variable Total_Income rather than simply x.

Always declare variables, even when using a language that does not require you to do so.

Prompt input and output so that it is user friendly.

Use blank lines to separate chunks of codes in a logical manner.

Break up large sections of code into Procedures or Functions whenever practical.

Follow standard guidelines and conventions for windows applications. This is especially important in the design of the graphical user interface of your application.

Indent code to show structures such as loops and decision statements. Indenting is used to visually improve the layout of a program. Indenting shows which lines are subordinate to other lines. For example, all the lines which make up the body of a loop should be indented as shown below ...

Example 1 - Indenting in a loop

```
for( int a = 10; a < 20; a = a + 1 )
{
    cout << "value of a: " << a << endl;
}
```

Example 2 - Indenting in IF-THEN-ELSE

```
if (num == 5)
{
        cout<<"You guessed right.\n";
}
else
{
        cout<<"You guessed wrong.\n";
}
```

Always include a File menu.

Always include an About command in the File menu (or Help) which contains information about the program (brief description, date of creation, etc.) and the author (name, school, class, teacher's name, etc.)

Include shortcut keys as you would see in a professional application.

Include ToolTip text where necessary.

Commenting Your Code

Comments are English statements that provide valuable information about your code.

In Visual C++, comments are preceded by //.

Multi-line comments start with /* and end with */

Comments are not processed by the Visual C++ compiler, so they will not make the compiled version of your program bigger.

Source: Problem Solving (First Ed.) J. Hume, (c)1993

Errors and Bugs

> A 2002 study funded by the National Institute of Standards and Technology estimated software errors cost the U.S. economy about $59.5 billion a year.
>
> "When Good Software Goes Bad"
> from CNN.com (August 8, 2003)
> by Jeordan Legon

A software bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from behaving as intended (ex. producing an incorrect or unexpected result). Most bugs arise from mistakes and errors made by people in either a program's source code or its design.

Souce: wikipedia.org

When writing computer programs, three types of errors can occur:

- Syntax errors
- Run-time errors
- Logic errors

Syntax errors

These are grammatical errors in the formulation of statements. Syntax errors include such things as misspelled commands and missing "."

Run-time errors

These are errors that cannot be detected until the program is running. The syntax of the statements is correct, but on execution they cause a situation to arise that results in a crash or an undefined value. Error handlers can be used to trap such errors and deal with them. Examples of run-time errors are attempted division by zero or trying to access a non-existent object.

Logic errors

These are errors that cause the program to behave incorrectly. They generally arise through failure on the part of the programmer to arrive at a correct algorithm for the task. Typical problems might be incorrect ordering of statements, failure to initialise or re-initialise a variable, assignment to an incorrect variable, use of '<' instead of '<=', use of 'and' instead of 'or', or omission of a crucial step in the processing. Logic errors may lurk in a program even when it appears to work - they may only surface under certain conditions. This is why careful testing is so important.

Result of Bugs

The results of bugs may be extremely serious.

- Bugs in the code controlling the Therac-25 radiation therapy machine were directly responsible for some patient deaths in the 1980s.
- In 1996, the European Space Agency's US$1 billion prototype Ariane 5 rocket was destroyed less than a minute after launch, due to a bug in the on-board guidance computer program.
- In June 1994, a Royal Air Force Chinook crashed into the Mull of Kintyre, killing 29. This was initially dismissed as pilot error, but an investigation by Computer Weekly uncovered sufficient evidence to convince a House of Lords inquiry that it may have been caused by a software bug in the aircraft's engine control computer.

Why are the called "Bugs"?

The story (which some people dispute the accuracy of) goes that the term was first used by computer pioneer Grace Hopper. Around 1947, Hopper was working on the Mark II and Mark III computers at Harvard Faculty in the Computation Laboratory. Operators traced an error in the Mark II to a moth trapped in a relay, and therefore called the problem a "bug". Hopper is also generally credited with first using the term "debug" in reference to fixing a problem with a computer program.

Introduction to Programming Exercises

_____ _____
Name Date

The User Interface (UI) of a Visual C++ program is created by placing Objects on a _____.

List the three types of errors you can encounter while programming.

_____

_____

_____

List five things you can do to make your code more readable. (Refer to: Language Syntax | Proper Programming Style)

_____

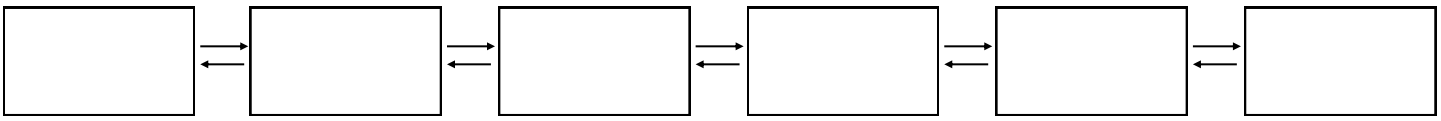
_____

_____

_____

_____

List the six steps of the Software Development Cycle.



List five Quality Requirements of computer programs.

_____

_____