Lesson 6 GETTING READY TO PROGRAM

Vocabulary List

algorithm, *n*
bug, *n*
C++ compiler
central processing unit (CPU)
code, *n*, *v*
compilation, *n*
compile, *v*
data, *n pl*
debug, *v*
debugger, *n*
debugging, *n*
digital camera
digital electronic machine
edit, *v*
error, *n*
executable file
execute, *v*
file, *n*, *v*
flowchart, *n*
input/output device
keyboard, *n*
linker, *n*
memory, *n*
object code
object file
operating system
preprocessor, *n*
printer, *n*
program, *n*, *v*
programmable, *adj*
programmer, *n*
programming, *n*
run, *n*, *v*
scanner, *n*
screen, *n*
source code
source file
store, *v*
test, *n*, *v*
testable, *adj*
tester, *n*
testing, *n*

A. Pre-Reading

– *Describe how you use computers in your study and free time.*
– *What programming language(s) do you already know / are you studying now / will you study at university later?*
– *When do we celebrate Programmer's day?*
– *Remember your first program. Tell your group mates about it.*

B. Reading

Getting Ready to Program

by Ira Pohl

Programs are written to instruct machines to carry out specific tasks or to solve specific problems. A step-by-step procedure that accomplishes a desired task is called an *algorithm*. Thus, *programming* is the activity of communicating algorithms to computers. We have all given instructions to someone in English and then had that person carry out the instructions. The programming process is analogous, except that machines have no tolerance for ambiguity and must have all steps specified in a precise language and in tedious detail.

The Programming Process

1. Specify the task.
2. Discover an algorithm for its solution.
3. Code the algorithm in C++.
4. Test the code.

A computer is a digital electronic machine composed of three main components: processor, memory, and input/output devices. The processor is also called the *central processing unit*, or *CPU*. The processor carries out instructions that are stored in the memory. Along with the instructions, data also is stored in memory. The processor typically is instructed to manipulate the data in some desired fashion. *Input/output devices* take information from agents external to the machine and provide information to those agents. Input devices are typically keyboards, mice, scanners, digital cameras, and joysticks. Output devices are typically screens, printers, speakers, and headphones. The physical makeup of a machine can be quite complicated, but the user need not be concerned with the details.

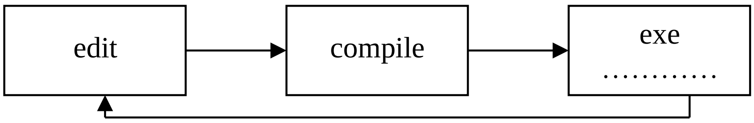
The *operating system* consists of a collection of special programs and has two main purposes. First, the operating system oversees and coordinates the resources of the machine as a whole. For example, when a file is created on a disk, the operating system takes care of the details of locating it in an appropriate place and keeping track of its name, size, and date of creation. Second, the operating system provides tools to users, many of which are useful to the C++ programmer. Two of these tools are of paramount importance: the text editor and the C++ compiler.

We assume the reader can use a text editor to create and modify files containing C++ code. C++ code is also called *source code*, and a file containing source code is called a *source file*. After a file containing source code (a program) has been created, the C++ compiler is invoked. This process is system-dependent. For example, on many UNIX systems, we can invoke the C++ compiler with the command

```
CC pgm.cpp
```

where *pgm.cpp* is the name of a file that contains a program. If there are no errors in *pgm.cpp*, this command produces an *executable file* – one that can be run, or executed. Although we think of this as compiling the program, what actually happens is more complicated.

When we compile a simple program, three separate actions occur: first the preprocessor is invoked, then the compiler, and finally the linker. The preprocessor modifies a copy of the source code by including other files and by making other changes. The compiler translates this into *object code*, which the linker then uses to produce the final executable file. A file that contains object code is called an *object file*. Object files, unlike source files, usually are not read by humans. When we speak of compiling a program, we really mean invoking the preprocessor, the compiler, and the linker. For a simple program, this is all done with a single command. After the programmer writes a program, it has to be compiled and tested. If modifications are needed, the source code has to be edited again. Thus, part of the programming process consists of this cycle:



When the programmer is satisfied with the program performance, the cycle ends.

C. Review Questions

Exercise 1. *Open Ended*

1. What is programming?
2. What are the steps in the development of a program?
3. How much time (in percents) does each step generally take?
4. What step(s) do **you** usually omit and why?
5. What do you call any collection of statements or declarations written in some human-readable computer programming language?
6. Look at the following program in C# and try to guess what it does without running it.

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int x = 1;
```

```
            for(int i = 1; i <= 10; i++)
            {
               x = x * 2;
               Console.WriteLine(x);
            }
        }
    }
```

7. What is the compiler and what function does it perform?
8. What is meant by a good coding style?
9. What are the qualities of a good program?

Real programmers code in binary

Exercise 2. True/False

1. Coding is writing a program.
2. A program can be defined as a set of written instructions created by a programmer or an executable piece of software.
3. Screens and printers are typical input devices.
4. The program has to be compiled and tested before it is written.
5. Your source code file is not a program, and it can't be executed, or run, as a program can.
6. To turn your source code into a program, you use a compiler.
7. After your source code is compiled, an object file is produced.
8. C++ code is also called object code.
9. A file that contains object code is called an executable file.
10. The files you create with your editor are called source files.
11. Whatever type of bug you find, you mustn't fix it.

Exercise 3. Word Building

Look at the groups of words and decide what part of speech each word is. Then complete the sentences with the correct word.

program programmer programming programmable

1. A …………………… earns £20, 000 a year.
2. A computer …………………… is a set of instructions that tells the computer what to do.
3. Converting an algorithm into a sequence of instructions in a programming language is called ……………………

compile compiler compilation

4. Programs written in a high-level language require ……………………, or translation into machine code.
5. A …………………… generates several low-level instructions for each source language statement.
6. Programmers usually …………………… their programs to create an object program and diagnose possible errors.

bug debug debugger debugging

7. It has been estimated that fully 90 percent of the cost of software is the combined cost of …………………… and maintenance.
8. There is *always* one more ……………☺.
9. The best compilers usually include an integrated …………………… which detects syntax errors.

tester test testing testable

10. *Gamma* …………………… is an informal phrase that refers ironically to the release of "buggy" products.
11. During the design phase, …………………… works with developers in determining what aspects of a design are …………………….
12. Modern operating systems usually …………………… the memory when the computer is switched on.