Structured Programming[*]

Kenneth Leroy Busbee

Abstract
An introduction to control structure categories and those allowed in structured programming.

1 Introduction

One of the most important concepts of programming is the ability to control a program so that different lines of code are executed or that some lines of code are executed many times. The mechanisms that allow us to control the flow of execution are called **control structures**. Flowcharting is a method of documenting (charting) the flow (or paths) that a program would execute. There are four main categories of control structures:

- **Sequence** – Very boring. Simply do one instruction then the next and the next. Just do them in a given sequence or in order listed. Most lines of code are this.

- **Selection** – This is where you select or choose between two or more flows. The choice is decided by asking some sort of question. The answer determines the path (or which lines of code) will be executed.

- **Iteration** – Also known as repetition, it allows some code (one to many lines) to be executed (or repeated) several times. The code might not be executed at all (repeat it zero times), executed a fixed number of times or executed indefinitely until some condition has been met. Also known as looping because the flowcharting shows the flow looping back to repeat the task.

- **Branching** – A control structure that allows the flow of execution to jump to a different part of the program. This category is rarely used in modular structured programming.

All high-level programming languages have control structures. All languages have the first three categories of control structures (sequence, selection, and iteration). Most have the **if then else** structure (which belongs to the selection category) and the **while** structure (which belongs to the iteration category). After these two basic structures there are usually language variations.

The concept of **structured programming** started in the late 1960's with an article by Edsger Dijkstra. He proposed a "go to less" method of planning programming logic that eliminated the need for the branching category of control structures. The topic was debated for about 20 years. But ultimately – "By the end of the 20th century nearly all computer scientists were convinced that it is useful to learn and apply the concepts of structured programming. "[1]

[*] Version 1.9: Mar 5, 2012 6:08 am -0600

[†] http://creativecommons.org/licenses/by/3.0/

[1] Structured programming from Wikipedia

1.1 Introduction to Selection Control Structures

The basic attribute of a selection control structure is to be able to select between two or more alternate paths. This is described as either **two-way selection** or **multiway selection**. A question using Boolean concepts usually controls which path is selected. All of the paths from a selection control structure join back up at the end of the control structure, before moving on to the next lines of code in a program.

We have mentioned that the **if then else** control structure belongs to the selection category and is a two-way selection.

Example 1: if then else control structure

```
if (age > 17)
  {
  cout ≪ "You can vote.";
  }
else
  {
  cout ≪ "You can't vote.";
  }
```

1.2 Introduction to Iteration Control Structures

The basic attribute of an iteration control structure is to be able to repeat some lines of code. The visual display of iteration creates a circular loop pattern when flowcharted, thus the word "loop" is associated with iteration control structures. Iteration can be accomplished with **test before loops**, **counting loops**, and **test after loops**. A question using Boolean concepts usually controls how long the loop will execute.

We have mentioned that the **while** control structure belongs to the iteration category and is a test before loop.

Example 2: while control structure

```
counter = 0;
while (counter < 5)
  {
  cout ≪ "\nI love computers!";
  counter ++;
  }
```

2 Definitions

Definition 1: control structures
Mechanisms that allow us to control the flow of execution within a program.

Definition 2: sequence
A control structure where you do the items in the sequence listed.

Definition 3: selection
A control structure where you select between two or more choices.

Definition 4: iteration
A control structure that allows some lines of code to be executed many times.

Definition 5: branching
A control structure that allows the flow of execution to jump to a different part of the program.

Definition 6: structured programming
A method of planning programs that avoids the branching category of control structures.