SWITCH STATEMENTS

CSC100 Introduction to programming in C/C++ (Spring 2023)

Marcus Birkenkrahe

April 24, 2023

Contents

1 README

- In this section of the course, we go beyond simple statements and turn to program flow and evaluation of logical conditions
- This section follows chapter 3 in Davenport/Vine (2015) and chapters 4 and 5 in King (2008)
- Practice workbooks, input files and PDF solution files in GitHub

2 Download the practice file

- Open the Emacs browser with `M-x eww` on `bit.ly/cc_switch`

- Write file with `C-x C-w` to `switch.org`
- Kill buffer with `C-x k`
- Re-open file with `C-x C-f`

3 Overview

- The `switch` statement is fairly complex: it combines conditional expressions, constant expressions, default and break statements.

```
switch ( expression ) {
  case constant expression : statements
  ...
  case constant expression : statements
  default : statements
}
```

- **Controlling expression:** an integer expression in parentheses, like `(5)`. Characters are treated as integers in C and cannot be tested, so `('a')` is not allowed.
- **Case labels:** each case begins with a constant expression, like `Case 5: -` this is like any other expression except that it cannot contain variables or function calls.
- **Statements:** any number of statements. No braces required around the statements. The last statement is usually `break` to close the case.

4 Simple example

- In the example code **??**, the `grade` is set in the variable declaration. Depending on the value, a `case` is triggered and the corresponding statements are executed.

☒ What is the output of **??** for 5,3,0,-1,0.5?

VALUE	OUTPUT
5	Failing
3	Passing
0	Illegal grade
-1	Illegal grade
0.5	Illegal grade

- The code:

```
int grade = 0.5;
switch (grade) {
 case 4:
 case 3:
 case 2:
 case 1:
   printf("Passing");
   break;
 case 5:
 case 6:
   printf("Failing");
   break;
 default:
   printf("Illegal grade");
   break;
 }

Illegal grade
```

☒ Which problem/solution set does the program implement?

> Answer: The program reflects "passing" grades 4,3,2,1, and "failing" grade 0. Any other grade value is not allowed. (This happens to be the European grade scale, which is A = 1 to D = 4, and F = 5 or 6.)

- You can also put several case labels on the same line as in **??**, which is otherwise identical to **??**.

```
int grade = 3;

switch (grade) {
 case 4: case 3: case 2: case 1:
   printf("Passing");
   break;
 case 5: case 6:
   printf("Failing");
   break;
```

```
 default:
   printf("Illegal grade");
   break;
 }

Passing
```

- Note: You cannot write a case label for a range of values.
- The default case (when none of the case expressions apply) is optional, and it does not have to come last.

5 The role of the `break` statement

- The `switch` statement is a *controlled jump*. The `case` label is a marker indicating a position within the switch.

☐ Let's run the previous program again, without the `break` statements. What do you think the output will be?

```
int grade = 5;

switch (grade) {
  // cases 4,3,2,1 all lead to a passing grade
 case 4:
 case 3:
 case 2:
 case 1:
   printf("Passing");
 case 5:
 case 6:
   printf("Failing");
 default:
   printf("Illegal grade");
 }

FailingIllegal grade
```

☐ What happens without the `break` statements?

Answer: When the last statement in a case has been executed, control falls through to the first statement in the following case; its case label is ignored. Without `break` (or some other jump statement, like `return` or `goto`, control flows from one case to the next.

- Deliberate falling through (omission of `break`) should be indicated with an explicit comment.

6 Let's practice!

- Open and complete the `switch.org` practice file that you downloaded
- Upload the completed file as an assignment to Canvas

7 References

- Davenport/Vine (2015) C Programming for the Absolute Beginner (3ed). Cengage Learning.
- GVSUmath (Aug 10, 2012). Proving Logical Equivalences without Truth Tables [video]. URL: youtu.be/iPbLzl2kMHA.
- Kernighan/Ritchie (1978). The C Programming Language (1st). Prentice Hall.
- King (2008). C Programming - A modern approach (2e). W A Norton.
- Orgmode.org (n.d.). 16 Working with Source Code [website]. URL: orgmode.org