Switch case in C

By Alex Allain

Switch case statements are a substitute for long if statements that compare a variable to several "integral" values ("integral" values are simply values that can be expressed as an integer, such as the value of a char). The basic format for using switch case is outlined below. The value of the variable given into switch is compared to the value following each of the cases, and when one value matches the value of the variable, the computer continues executing the program from that point.

```
switch ( <variable> ) {
case this-value:
  Code to execute if <variable> == this-value
  break;
case that-value:
  Code to execute if <variable> == that-value
  break;
...
default:
  Code to execute if <variable> does not equal the value following any of the cases
  break;
}
```

The condition of a switch statement is a value. The case says that if it has the value of whatever is after that case then do whatever follows the colon. The break is used to break out of the case statements. Break is a keyword that breaks out of the code block, usually surrounded by braces, which it is in. In this case, break prevents the program from falling through and executing the code in all the other case statements. An important thing to note about the switch statement is that the case values may only be constant integral expressions. Sadly, it isn't legal to use case like this:

```
int a = 10;
int b = 10;
int c = 20;

switch ( a ) {
case b:
  /* Code */
  break;
case c:
  /* Code */
  break;
default:
  /* Code */
  break;
}
```

The default case is optional, but it is wise to include it as it handles any unexpected cases. It can be useful to put some kind of output to alert you to the code entering the default case if you don't expect it to. Switch statements serve as a simple way to write long if statements when the requirements are met. Often it can be used to process input from a user.

Below is a sample program, in which not all of the proper functions are actually declared, but which shows how one would use switch in a program.

```
#include <stdio.h>

void playgame()
{
    printf( "Play game called" );
```

```
{
    printf( "Play multiplayer game called" );
}

int main()
{
    int input;

    printf( "1. Play game\n" );
    printf( "2. Load game\n" );
    printf( "3. Play multiplayer\n" );
    printf( "4. Exit\n" );
    printf( "Selection: " );
    scanf( "%d", &input );
    switch ( input ) {
        case 1:             /* Note the colon, not a semicolon */
            playgame();
            break;
        case 2:
            loadgame();
            break;
        case 3:
            playmultiplayer();
            break;
        case 4:
            printf( "Thanks for playing!\n" );
            break;
        default:
            printf( "Bad input, quitting!\n" );
            break;
    }
    getchar();

}
```

This program will compile, but cannot be run until the undefined functions are given bodies, but it serves as a model (albeit simple) for processing input. If you do not understand this then try mentally putting in if statements for the case statements. Default simply skips out of the switch case construction and allows the program to terminate naturally. If you do not like that, then you can make a loop around the whole thing to have it wait for valid input. You could easily make a few small functions if you wish to test the code.

Quiz yourself
Previous: Functions
Next: Pointers
Back to C Tutorial Index