

Switch Statement Using Bloodshed Dev C++

By Janine Bouyssounouse

Bloodshed Dev C++ is a free program to make writing and compiling C++ programs easy to do. It makes executable files quickly so programs can be shared with others as soon as they are written and debugged. Bloodshed Dev C++ can be downloaded from the website:

<http://www.bloodshed.net/devcpp.html>.

There is more than one way to start writing programs in Dev C++. This document discusses using a project for each program. This encourages using an organized file structure to keep each program in a separate location for future use. Remember that reusing code is a good idea and it is nice to be able to find the folder with everything for a program in one place.

Click on the File menu and select New – Project. Choose Console Application for this program. Type in a name for the project, such as Display Input, in the name field in the bottom left corner of the window. Click on the OK button. Choose a place on the computer to save the project. It's a good idea to make a folder for each project. Once the project is saved, some basic items show up on the screen. These items give a shell of a C++ program and include things that are unique to Bloodshed Dev C++ as well.

```
#include <iostream>    - This tells the compiler to include files.
#include <stdlib.h>     - This tells the compiler to include files.

using namespace std;  - This tells the compiler that you will be using key
                      words that are included in namespace std.

int main(int argc, char *argv[])    - This starts the main function.
{                                     - Main is contained between { and }.

    system("PAUSE");                - This leaves the console window open until
                                     you are ready to close it. Otherwise you
                                     wouldn't be able to see the results of the
                                     program. Not all compilers require this.

    return 0;                        - This is how the main program ends.
}                                     - This signifies the end of the code for main.
```

The code for the main part of the program will be typed between the first curly brace ({} and the system("PAUSE") line of code. The input and print functions for this program will be outside of this area, but the main program will refer to them.

Programs have a need to do different things depending on user input or based on the flow of the program itself. If statements can do this work, but sometimes it is much easier to use other forms that might work more easily for menu items or what to do with different rolls of a die. The switch statement can be very helpful in handling multiple choices in a program.

This program will ask the user to select a number from a predetermined set of numbers. Then the program will print different things on the screen depending on which number the user chose.

Type in the first lines of the program by clicking at the beginning of the first line and pressing enter a couple of times, then moving to the first blank line. Type in these first lines of code:

```
// This program uses a switch statement to print different things to the  
// screen based on user input.
```

The next line of code can be a comment stating your name as the programmer and the date the program was written.

```
// Written by Janine Bouyssounouse on 01/21/09
```

Type the following line of code after the using namespace statement:

```
void welcome();
```

A welcome statement will be displayed to explain what this program does. There is nothing sent to or received from the welcome function, so the void and empty parentheses are used in the function prototype.

On the last line of the program, we will comment the line to show that the main function is finished, so that it is not confused with the other functions listed after it.

The last line of code should look like this:

```
} // end main
```

Next we will start typing the welcome function at the end of the program, outside of the curly braces for the main function.

Skip a line and type:

```
// welcome function displays an opening message to  
// explain the program to the user  
void welcome()
```

Notice the comments are listed on the lines before the start of the function. The first line of the function looks exactly like the function prototype, except for the missing semicolon at the end.

On the next line of code, type in the function:

```
{  
  cout << "This program displays different messages depending\n";  
  cout << "on which number is entered by the user.\n";  
  cout << "Pick a number between 1 and 6 to see what\n";  
  cout << "the program will say.\n\n";  
} // end of welcome function
```

This function uses no variables at all.

Call the welcome function in the main function to display the information to the user.

To do this, type the following code after the first curly brace in the main function and before the `system("PAUSE")` line of code:

```
welcome(); // This calls the welcome function
```

We are going to get an integer from the user and send that integer to the display function which will use the switch statement to decide what to print on the screen. First we need a function to get the integer.

Type the following line of code after the `void welcome();` statement in the function prototype section before the main function:

```
int getInteger();
```

`int` means the function will return an integer value from the function. `getInteger` is the name of the function. The empty parentheses show that nothing is being passed to the `getInteger` function from the main program. The semicolon shows the end of the line of code for the function prototype.

At the end of the program, type in the `getInteger` function:

```
// getInteger asks the user for a number between 1 and 6.
// The integer is returned to where the function was called.
int getInteger()
{
    int response; // declares variable called response

    cout << "Please type a number between 1 and 6: "; // prompt for number
    cin >> response; // gets input from user and assigns it to response

    return response; // sends back the response value
} // end getInteger function
```

Since the integer is being sent back to the main function, an integer needs to be declared in the main function to receive the value from the function. Type the following line of code in the main function before the call to the `welcome` function:

```
int choice; // declares the choice variable
```

Now call the `getInteger` function from the main function. Type the following code after the `welcome` function is called in the main function:

```
choice = getInteger(); // calls getInteger and receives the value for choice
```

Another function needs to be created for the different statements to be printed to the screen based on the user input. This is where the `switch`

statement will be. This will be a void function (nothing is returned) that will receive the choice variable when it is called. Type the following line of code at the end of the list of function prototypes before the main function:

```
void displayResponse(int choice);
```

This function returns a character (char) which will be the y or n answer to the question that is sent to the function (string question).

Now type the function at the very bottom of the program:

```
// displayResponse function takes the int variable and uses it
// to determine which set of tasks will be performed.
void displayResponse(int choice)
{
    // switch statement based on the choice variable
    switch (choice) // notice no semicolon
    {
        case 1: // choice was the number 1
            cout << "One is a lonely number and very useful in math.\n";
            break; // this ends the statements for case 1

        case 2: // choice was the number 2
            cout << "Two is the only even prime number.\n";
            break; // this ends the statements for case 2

        case 3: // choice was the number 3
            cout << "Three is a crowd and also a prime number.\n";
            break; // this ends the statements for case 3

        case 4: // choice was the number 4
            cout << "Four square is a fun game to play, but four squared is ";
            cout << 4 * 4 << "\n";
            break; // this ends the statements for case 4

        case 5: // choice was the number 5
            cout << "Counting by fives is fun. Five, Ten, Fifteen, Twenty...\n";
            break; // this ends the statements for case 5
    }
}
```

```
case 6: // choice was the number 6
    cout << "Six is divisible by two and three.\n";
    break; // this ends the statements for case 6

default: // used when choice falls out of the cases covered above
    cout << "You didn't pick a number between 1 and 6.\n";
    again = getInteger(); // gives the user another try
    displayResponse(again); // recalls displayResponse with new number
    break;
} // end of switch statement
} // end displayResponse function
```

The switch statement has many parts to it. The cases that are handled by the switch statement each have their own lines of code and are only executed if the specific case is needed. Case 3 is only executed if choice is three. A default case is important to catch anything that falls outside of the options handled in the case statements.

Notice the two functions being called again from within this function. The program continues to ask for another number until the user types in an acceptable number. This is one way to ensure the correct number is entered by the user. There are other ways to handle the situation.

Now the function is ready to be called from the main function. Type the following line of code after the getInteger function is called in the main function:

```
displayResponse(choice); // passes choice to displayResponse function
```

Nothing is returned from this function and the program ends. Here is the code for the finished program:

```
// This program uses a switch statement to print different things to the
// screen based on user input.
// Written by Janine Bouyssounouse on 01/21/09

#include <iostream>
#include <stdlib.h>
```

```

using namespace std;
void welcome();
int getInteger();
void displayResponse(int choice);

int main(int argc, char *argv[])
{
    int choice; // declares the choice variable

    welcome(); // This calls the welcome function
    choice = getInteger(); // calls getInteger and receives the value for choice
    displayResponse(choice); // passes choice to displayResponse function

    system("PAUSE");
    return 0;
} // end main

// welcome function displays an opening message to
// explain the program to the user
void welcome()
{
    cout << "This program displays different messages depending\n";
    cout << "on which number is entered by the user.\n";
    cout << "Pick a number between 1 and 6 to see what\n";
    cout << "the program will say.\n\n";
} // end of welcome function

// getInteger asks the user for a number between 1 and 6.
// The integer is returned to where the function was called.
int getInteger()
{
    int response; // declares variable called response

    cout << "Please type a number between 1 and 6: "; // prompt for number
    cin >> response; // gets input from user and assigns it to response

    return response; // sends back the response value
} // end getInteger function

// displayResponse function takes the int variable and uses it

```

```

// to determine which set of tasks will be performed.
void displayResponse(int choice)
{
    int again;

    // switch statement based on the choice variable
    switch (choice) // notice no semicolon
    {
        case 1: // choice was the number 1
            cout << "One is a lonely number and very useful in math.\n\n";
            break; // this ends the statements for case 1

        case 2: // choice was the number 2
            cout << "Two is the only even prime number.\n\n";
            break; // this ends the statements for case 2

        case 3: // choice was the number 3
            cout << "Three is a crowd and also a prime number.\n\n";
            break; // this ends the statements for case 3

        case 4: // choice was the number 4
            cout << "Four square is a fun game to play, but four squared is ";
            cout << 4 * 4 << ".\n\n";
            break; // this ends the statements for case 4

        case 5: // choice was the number 5
            cout << "Counting by fives is fun. Five, Ten, Fifteen, Twenty...\n\n";
            break; // this ends the statements for case 5

        case 6: // choice was the number 6
            cout << "Six is divisible by two and three.\n\n";
            break; // this ends the statements for case 6

        default: // used when choice falls out of the cases covered above
            cout << "You didn't pick a number between 1 and 6.\n\n";
            again = getInteger(); // gives the user another try
            displayResponse(again); // recalls displayResponse with new number
            break;
    } // end of switch statement
} // end displayResponse function

```

Save, compile and run the program to see if it works. Choose Compile and Run from the Execute menu.

Here is a display of the program:

This program displays different messages depending on which number is entered by the user.

Pick a number between 1 and 6 to see what the program will say.

Please type a number between 1 and 6: 8
You didn't pick a number between 1 and 6.

Please type a number between 1 and 6: 9
You didn't pick a number between 1 and 6.

Please type a number between 1 and 6: 2
Two is the only even prime number.

Press any key to continue . . .

Now write a program of your own.

Exercise 1: Write a program using a switch statement to handle user input of either a, b or c. Remember to include the upper case as well as the lower case letters in the case structure.

Example:

```
case 'A':  
case 'a':  
    cout << "You picked a!\n";  
    break;
```

Sample Code for Exercise 1:

```
// This program uses a switch statement to print different things to the
// screen based on user input.
// Written by Janine Bouyssounouse on 01/21/09

#include <iostream>
#include <stdlib.h>

using namespace std;
void welcome();
char getChar();
void displayResponse(char choice);

int main(int argc, char *argv[])
{
    char choice; // declares the choice variable

    welcome(); // This calls the welcome function
    choice = getChar(); // calls getChar and returns the value for choice
    displayResponse(choice); // passes choice to displayResponse function

    system("PAUSE");
    return 0;
} // end main

// welcome function displays an opening message to
// explain the program to the user
void welcome()
{
    cout << "This program displays different messages depending\n";
    cout << "on which letter is entered by the user.\n";
    cout << "Pick a letter a, b or c to see what\n";
    cout << "the program will say.\n\n";
} // end of welcome function

// getChar asks the user for a letter a, b or c.
// The character is returned to where the function was called.
char getChar()
{
```

```

char response; // declares variable called response

cout << "Please type a letter a, b or c: "; // prompt for letter
cin >> response; // gets input from user and assigns it to response

return response; // sends back the response value
} // end getChar function

// displayResponse function takes the char variable and uses it
// to determine which set of tasks will be performed.
void displayResponse(char choice)
{
    char again;

    // switch statement based on the choice variable
    switch (choice) // notice no semicolon
    {
        case 'A': // choice was the letter A
        case 'a': // choice was the letter a
            cout << "A is for apple.\n\n";
            break; // this ends the statements for case A/a

        case 'B': // choice was the letter b
        case 'b': // choice was the letter b
            cout << "B is for baseball.\n\n";
            break; // this ends the statements for case B/b

        case 'C': // choice was the letter C
        case 'c': // choice was the letter c
            cout << "C is for cat.\n\n";
            break; // this ends the statements for case C/c

        default: // used when choice falls out of the cases covered above
            cout << "You didn't pick a letter a, b or c.\n\n";
            again = getChar(); // gives the user another try
            displayResponse(again); // recalls displayResponse with new character
            break;
    } // end of switch statement
} // end displayResponse function

```

Display from Sample Code for Exercise 1:

This program displays different messages depending on which letter is entered by the user.

Pick a letter a, b or c to see what the program will say.

Please type a letter a, b or c: y

You didn't pick a letter a, b or c.

Please type a letter a, b or c: b

B is for baseball.

Press any key to continue . . .