# Original Code

In this section, images of the original code loaded into the XCode IDE are featured in figures 1.1 – 1.3 and the code being compiled, and run is featured in figures 2.1 – 2.3. Once compiled and run, the code outputs a menu of possible choices to the terminal as seen in figure 2.1. As seen in figure 2.2, upon the user entering a “B” the program executes the correct printf() command which prints, “Welcome to baseball season!” and then prints the menu of choices again. Because the Original code lacks break statements for every case in the switch case break statement, it then prints the default statement of, “Please enter a valid selection” and then prints the menu of choices again. Finally, as seen in figure 2.3, the user enters, “F”, which prints to the terminal, “Welcome to Football season!” which is the correct response but then it incorrectly prints “Welcome to Soccer season!” because the lack of a break statement corresponding to the “F” or “f” cases allows the following case of “S” or “s” to execute.

### Figure 1.1. Image 1 of 3 of the original code in the XCode IDE.

Text

Description automatically generated

### Figure 1.2. Image 2 of 3 of the original code in the XCode IDE.

Text

Description automatically generated

### Figure 1.3. Image 3 of 3 of the original code in the XCode IDE.

Text

Description automatically generated

### Figure 2.1. Image 1 of 3 of the original code being compiled and run in the XCode IDE.

Text

Description automatically generated

### Figure 2.2. Image 2 of 3 of the original code being compiled and run in the XCode IDE with the user interaction of entering “B”.

Text

Description automatically generated

### Figure 2.3. Image 3 of 3 of the original code being compiled and run in the XCode IDE with the user interaction of entering “F”.

Text

Description automatically generated

# Refactoring the code for proper execution

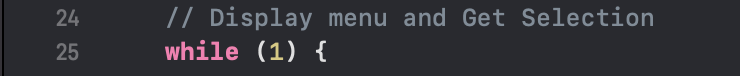
The first thing that was fixed was making sure that each case in the switch statement inside showResults() has a break statement. Initially, the “B” and “b” cases had no break statement after the printf() statement. This caused the program to “fall through” to the next case every time the “B” or “b” were executed as can be seen in figure 2.3. The default case does not need a break statement so that was left alone. After that, the main function was edited to avoid the double printing of the greeting and menu as can be seen happening in figures 2.2 and 2.3 when “B”, “b”, “E”, “e”, “S”, or “s” are entered. This was corrected by modifying the while loop in the main() function to check if the character received by getchar() is “\n”. This is done using an if suite pictured in figure 4. For the sake of simplicity, the while loop expression was changed from “(cont != ‘E’ && cont != ‘e’)” to “(1)” as seen in figure 5. The while loop is still only exited if cont is equal to ‘E’ or ‘e’ but that is now controlled by an if suite seen in figure 6.

### Figure 4. Image of the while loop that calls showMenu(), showResults(), and getchar() which includes the if suite that keeps the menu from being double printed by checking cont for “\n”.

Text

Description automatically generated

### Figure 5. Image of the while loop expression that controls the calls to showResults(), showMenu(), and getchar().



### Figure 6. Image of the if suite that will execute a break statement in if cont is equal to ‘E’ or ‘e’.

Graphical user interface

Description automatically generated

## MSC17-C. Finish every set of statements with a case label with a break statement.