

## Learning Activity

Converting a do-while statement to a while statement, in the event you need a reminder.

### Existing Code

```
        string quit;
        do
    {
        try
            {
                // retrieve user input and display miles per gallon
                Console.Write( "Enter miles driven: " );
                double miles = Convert.ToDouble( Console.ReadLine() );
                Console.Write( "Enter gallons used: " );
                double gallons = Convert.ToDouble( Console.ReadLine() );
                Console.WriteLine( "{0:F} miles per gallon.",
                    CalculateMPG( miles, gallons ) );
            } // end try
        catch ( FormatException formatExceptionParameter )
        {
            Console.WriteLine( "Please enter decimal numbers for " +
                "the miles driven and gallons used." );
        } // end catch

        Console.Write( "Quit (yes/no)? " );
        quit = Console.ReadLine().ToLower();
    } while ( quit != "yes" );
```

As you know, a do-while statement allows an event to occur once, whether it was supposed to happen or not. If, by testing at the end (bottom) of the statement, one realizes the event was not qualified to happen, it's too late. Example: I write a program to take a lot of your money out of your savings account. A do-while would permit the extraction if "string quit;" was set to "string yes;" and my code took your money without challenge.

A while statement, on the other hand, prompts you (kinda asks) if I can take your money, and if testing at the top, and your answer is "no", the code to withdraw your funds is never processed. Sad for me, good for you.

Remember the premise and structure of a while statement.

1. Prompt the User for a response.
2. Obtain the User's input to the prompt.
3. Test the User's input against the while statement parameters in the argument (parenthesis). If true, the while statement is processed. If not true (false), then the code in the while statement is never activated.
4. If true and while in the while statement code, at the end of its processing, I have to prompt the user, then obtain the User's input in order to determine if the while statement remains active or not; that is, terminate and exit the while.

## Possible While Code

### The first prompt outside of the while ...

```
Console.Write("    1. Enter the number of miles \n        driven or a zero to exit  
->    ");
```

### Obtain the user's input ...

```
miles = Convert.ToDouble(Console.ReadLine());
```

### The while ...

```
while(miles != 0)
{
    Console.Write("    2. Enter the gallons used    ->    ");
    double gallons = Convert.ToDouble( Console.ReadLine() );
    Console.WriteLine("        -----");
    Console.WriteLine("        Miles per Gallon        ->    {0:F}\n",
        CalculateMPG( miles, gallons ) );

    // Successful accumulators
    totalMilesDriven += miles;
    totalGallonsUsed += gallons;
    numberOfValidEntries++;

    Console.Write("    1. Enter the number of miles \n        driven or a zero to exit    ->    ");
    miles = Convert.ToDouble(Console.ReadLine());

    Console.Write("\n\n    1. Enter the number of miles\n        driven or a zero to exit    ->    ");
    miles = Convert.ToDouble(Console.ReadLine());
}
// end WHILE
```

The Last two lines of code just before the while ends is the re-prompt and obtaining the user input, in order to determine if the while is to continue or not.

Ah, but, you already know all that.