Catching run-time errors

When you are able to predict potential runtime errors, by considering all eventualities, you can provide code to handle each Exception class error that may arise – by adding a try-catch construct. Your program can supply information to the user about the error, should you wish to do so, then proceed normally:

1. Add this program code to request user input of two numeric values for addition, then display their sum total
   ```csharp
   Console.Write( "Please Enter A Number: " );
   double num1 = Convert.ToInt16( Console.ReadLine( ) ) ;
   
   Console.Write( "Now Enter Another Number: " );
   double num2 = Convert.ToInt16( Console.ReadLine( ) ) ;
   
   Console.WriteLine( "Total: " + ( num1 + num2 ) ) ;
   ```

2. Press Start or F5 to run the application, then enter any six-figure integer and hit Enter

   ![Image of compiler reporting an OverflowException error]

3. The compiler reports an OverflowException error

   ![Image of compiler error message]

An int16 is a 16-bit integer within the range -32,768 to +32,767 – whereas an int32 is a 32-bit integer within -2,147,483,648 to +2,147,483,647.

4. Click the Stop Debugging button so you can edit the code