```
1  #include <stdio.h>
2  /*------------------------------------------------------------------------*/
3  void main(void)
4  {
5      printf("Here's one way to print a ");
6      printf("long string.\n");
7
8      printf("Here's another way to print a \
9  long string.\n"); // Type \ then Shift + Enter
10
11     printf("Here's the newest way to print a"
12         " long string. \n");
13 }
14 /*------------------------------------------------------------------------*/
15 //***DOS Console Program Output***
16 /*
17 Here's one way to print a long string.
18 Here's another way to print a long string.
19 Here's the newest way to print a long string.
20
21 C:\Users\KWhite\source\repos\Beginning Code\Debug\Beginning Code.exe (process
22 13408) exited with code 0.
23 Press any key to close this window . . .
24
25 */
26
```

Commented [KW1]: Preprocessor Directive Header file that contains the function definition for printf().

Commented [KW2]: Every C program must contain the main() function.

Commented [KW3]: The printf() function is used to print text on the screen in the form of a string. The string text is bounded by " " and is contained inside the parenthesis and terminated with a semicolon.

Code lines 5 and 6 are outputted on Line 17.

Commented [KW4]: Various ways to accomplish the same task as lines 5 and 6.

Code lines 8 and 9 are outputted on line 18 and code lines 11 and 12 are outputted on line 19.

Commented [KW5]: Start of the comment section where program output is pasted. Comments are ignored by the Compiler.

Commented [KW6]: Directory where the executable file was found and the termination code showing it executed correctly.

```
1  #include <stdio.h>
2  /*------------------------------------------------------------------------*/
3  void main (void)
4  {
5    int iNum;
6
7    iNum = 1;
8
9    printf("I am a simple computer. \n");
10   printf("   My favorite number is %d because it is the first. \n", iNum);
11 }
12 /*------------------------------------------------------------------------*/
13 //***DOS Console Program Output***
14 /*
15 I am a simple computer.
16    My favorite number is 1 because it is the first.
17 */
18
```

Commented [KW7]: Creating a programmer variable named iNum that is defined/declared as an integer.

Commented [KW8]: iNum is assigned the integer value of 1. The = sign is called the assignment operator which means data from the right hand side of it is stored on the left hand side of it.

Commented [KW9]: Notice there are three spaces after the first " which get printed on line 16. Also notice the %d inside the set of double quotes, the comma operator after the ending double quote, and the variable name iNum following the comma operator. The %d is a format specifier and placeholder where the data stored in iNum is displayed. Only data inside of the double quotes gets displayed.

```
1  #include <stdio.h>
2  #pragma warning(disable:4996)
3  /*-------------------------------------------------------------------------*/
4  void main (void)
5  {
6        int iWeight;
7        float fValue_per_lb, fValue;
8
9        printf("Enter your value in dollars per pound => ");
10       scanf("%f", &fValue_per_lb);
11
12       printf("\nEnter your weight in pounds => ");
13       scanf("%d", &iWeight);
14
15       fValue = iWeight * fValue_per_lb;
16
17       printf("\nYou are worth %f dollars.\n", fValue);
18       printf("You are worth %0.2f dollars. \n", fValue);
19       printf("You are worth %8.2f dollars. \n", fValue);
20       printf("You are worth $%0.2f.  \n", fValue);
21       printf("You are worth $%0.2e.  \n", fValue);
22 }
23 /*-------------------------------------------------------------------------*/
24 //***DOS Console Program Output***
25 /*
26 Enter your value in dollars per pound => 25
27
28 Enter your weight in pounds => 200
29
30 You are worth 5000.000000 dollars.
31 You are worth 5000.00 dollars.
32 You are worth  5000.00 dollars.
33 You are worth $5000.00.
34 You are worth $5.00e+03.
35
36 C:\Users\KWhite\source\repos\Beginning Code\Debug\Beginning Code.exe (process
37 15876) exited with code 0.
38 Press any key to close this window . . .
39 */
40
```

Commented [KW10]: This Preprocessor Directive is needed use the scanf() function found in stdio.h – a Microsoft requirement.

Commented [KW11]: Create program variables of data types integer and float. Notice the standard convention is to use i as the first character for integer data and f as the first character for float data. Also note the use of the _ when creating variable names of multiple words.

Commented [KW12]: The scanf() function takes keyboard input and stores it in a variable. The %f inside of the double quotes tells the function the type of data it will be receiving and storing. The comma operator separates the fields like it does with the printf(). The & immediately preceding the variable name is the address of operator and is required when using the variable name. Source code lines 9 and 10 are displayed on line 26.

Commented [KW13]: A C statement that gets the data stored in iWeight and fValue_per_lb, multiplies the data together, and then stores it in fVlaue.

Commented [KW14]: The same data can be represented numerous ways using various format specifiers. Notice the default is %f which displays everything to the left of the decimal point and 6 digits to the right. The %8.2f is defined as 8 for the minimum size for total character field width, with 2 characters to the right of the decimal point. – see line 32 and notice the extra space in front of the 5 making the field size = 8.

```
#include <stdio.h>
#pragma warning(disable:4996)
/*------------------------------------------------------------------------*/
void main(void)
{
	int iNum;

	printf("Enter integer value 38 => ");
	scanf("%d", &iNum);

	printf("\nInteger value 38 = %d in base 10.\n", iNum);
	printf("Integer value 38 = %o in base 8.\n", iNum);
	printf("Integer value 38 = %x in base 16.\n", iNum);
	printf("Integer value 38 = %c as an ASCII character.\n", iNum);
	printf("Integer value 38 = %f when viewed as a float.\n", iNum);
}
/*------------------------------------------------------------------------*/
//***DOS Console Program Output***
/*
Enter integer value 38 => 38

Integer value 38 = 38 in base 10.
Integer value 38 = 46 in base 8.
Integer value 38 = 26 in base 16.
Integer value 38 = & as an ASCII character.
Integer value 38 = 0.000000 when viewed as a float.

C:\Users\KWhite\source\repos\Beginning Code\Debug\Beginning Code.exe (process
8568) exited with code 0.
Press any key to close this window . . .

*/
```

Commented [KW15]: Display the base 10 data that is equal to 38 and stored in iNum as various other data type outputs. This data is actually stored as binary within the computer memory but can be displayed as various other forms when specified by the programmer.

Commented [KW16]: Results of integer number 38 when displayed in various formats. The details of the conversions will be investigated in next weeks theory and assignment folders. For now, just be aware that any number has multiple programmer options that may or may not produce the desired results when displayed.