

# Programming Fundamentals

## CS1336

### Assignment #1

#### Assignment #1 – Basic Programming

##### Introduction

---

Your first programming assignment will consist of two small C++ programs. Each program will be independent of the other program. Each one should compile correctly and produce the specified output. In addition, please follow the naming conventions for source code files as described in the file “Formatting Notes on Assignments.”

Note that each of the programs should comply with the commenting and formatting rules we discussed in class. For example, there should be a header for the whole program that gives the author’s name, class name, date, and description. End braces should be commented, and there are alignment and indenting requirements as discussed. Please ask if you have any questions.

##### Program #1

---

Please write a working C++ program that prints the following items to the console. (You can do all of these in one program; it is not necessary to write separate programs for each item.)

1. “Hello World!” (using an `endl` to put a carriage return at the end of the line)
2. “C++ Programming is fun!\n” (without using an `endl` to put a carriage return at the end of the line. As you will see, the `’\n’` character does the same thing when it is embedded inside a string.)
3. A prompt that says “The value of 2 times 3 is: ” followed by that calculated value. That can be accomplished with the following statement:

```
cout << "The value of 2 times 3 is: " << (2*3) << endl;
```

(NOTE: As illustrated here, don’t forget to use parenthesis around any calculations that are embedded in an output stream.)

4. The following diagrams, just as they appear here (side by side, not vertical):

NOTE: Please do not use loops for this part of the assignment. This is an exercise in manually formatting output only.

Create these diagrams exactly as they are listed here. For example, don’t create the rectangle, arrow, and diamond separately or vertically. They should all be on the same level in the diagram.

```

*****      *      *
*          *    ***      *  *
*          *  *****      *  *
*          *    *        *    *
*          *    *        *    *
*          *    *        *    *
*          *    *        *    *
*          *    *        *    *
*          *    *        *    *
*****      *      *

```

## Program #2

---

Please write a working C++ program that implements the following pseudocode. (Note: this project is based on the formula  $S = \left(\frac{1}{2}\right)at^2$ , which calculates the distance traveled under acceleration “a” and time “t”.)

*Create a variable of type float named timeVal and store the value 5.36 in it.*

*Create a variable of type float named acceleration and store the value 32.0 in it.*

*Create a variable of type double named distanceVal.*

*Calculate the value S according to the formula  $S = (1.0 / 2) * a * t^2$  and store that result in the variable distanceVal.*

*Display the results as follows:*

```

The time was:                5.4 sec.
The acceleration was:        32.0 feet/sec/sec
The distance traveled was:    XXXX.X feet

```

Of course, the “XXXX.X” value should be replaced by the calculated number. Note that the results should all line up as illustrated here, and that each number should be followed by the given units.

**NOTE:** In C++, controlling the number of decimal places that are displayed when floating point type variables are printed requires the use of “stream manipulators” that we haven’t yet covered. If you put the following `#include` statement at the top of your program:

```
#include <iomanip>
```

and include the following `cout` statement immediately prior to the first output statement, it should take care of the problem for now:

```
cout << fixed << setprecision(1);
```

This statement only needs to be executed one time. It will force C++ to display one decimal point when floating point numbers are printed. We’ll discuss this issue in more detail when we get to it in lecture.