

Tutorial/Lab | Week 02

Overview

The week 02 tutorial/lab is for you to practice and become familiar with the basics of using C++. You will be familiar with a number of these features from your Java programming experiences.

Tutorial Questions

Look at the C++ program below, and answer the questions.

```
1  #include <stdio>
2  #include <iostream>
3
4  #define EXIT_SUCCESS    0
5  #define LENGTH          10
6
7  double foo(int x, float y, char z);
8
9  int main (void) {
10     int i;
11     float f;
12     char c;
13     double d;
14     int array[LENGTH] = {10, 9, 8};
15
16     std::cout << "foo:\t" << d << "+" << f << "*" << c << "=" << d << std::endl;
17     printf("foo:\t%d + %.2f * %c = %.2lf\n", i, f, c, d);
18
19     for (int j = 0; j < LENGTH; j++) {
20         std::cout << array[j] << std::endl;
21     }
22
23     std::cin >> i;
24     std::cin >> f;
25     std::cin >> c;
26     d = foo(i, f, c);
27
28     return EXIT_SUCCESS;
29 }
30
31 double foo(int x, float y, char z) {
32     return x + y * z;
33 }
```

1. What is the purpose of lines 1 & 2? Name other standard header files.
2. Which function is called at program start-up? What value should this function return?
3. The example uses a variety of data types, what are they and how do they differ?
4. What is wrong with lines 10 - 13?
5. What is the use of the `#define` statements in lines 4 & 5?
6. What is the difference between the use of `cout` and `printf`?
7. What will lines 19 - 21 output?
8. How does reading from input in lines 23-25 work?
9. How are parameters passed to the function `foo` on line 26?
10. What happens in line 32?

Lab Questions

This lab contains a series of exercises. You may have done many of these exercises in Java already, however, it is good to practice these same exercises to become familiar with writing and compiling C++ programs.

It is a good idea to attempt the lab questions before coming to class. The lab might also be longer than you can complete in 2 hours. It is a good to finish the lab at home.

You should demonstrate your work to your tutor.

Unix Proficiency

The first thing to for this lab is to make sure that you are familiar with using Unix. The Unix Survival Guide is linked from the Week 01 module. If you have not already done so, work through the survival guide.

Exercises

1. Write two C++ programs that prints out **Hello World!**. Your implementations should:
 - (a) Use a `#define` to create an appropriate return value for the `main` function
 - (b) One implementation should use `cout` and the other should use `printf`.
 - (c) The implementations should only include the necessary header files.
2. Write a series of C++ programs (or one single program) to output the following star patterns. You should use basic control structures, such as `if`, `while` and/or `for` statements to output these patterns, rather than using fully expanded output statements.

(a) *

```
**
***
****
*****
*****
```

(b)

```
  *
  **
 ***
****
*****
*****
```

(c)

```
  *
 ***
*****
*****
*****
*****
*****
```

(d)

```
  *
 ***
*****
*****
*****
*****
*****
*****
*****
*****
***
  *
```

```
(e) * * * * *
    * * * * *
    * * * * *
    * * * * *
    * * * * *
    * * * * *
    * * * * *
```

3. Write a C++ program to read in information from a person (name, age, date of birth) and display it back to them. You may collect the information however you see fit.

```
Name: Gary Gygax
Age: 79
DOB: 27/07/1938
```

Answer the following questions:

- What problems did you encounter when reading in information?
 - Did you use `cin` or `scanf` to get the information and why?
4. Write a C++ program that does the following in order:
- reads in 10 integers
 - stores the integers in an array
 - sums the array
 - prints out the result.
5. Write a C++ program, that reads from the user a radius for a circle and prints out the area of the circle. This program should:
- Read in the radius as a `float` or `double`
 - Use a *function* to compute the area of the circle. You should define the parameters and return value of the function as you see fit. You should use a `#define` for the value of Pi (π).
 - Use a *function prototype* to *declare* the function.
6. The following infinite series can be used to calculate the value of π .

$$\pi = 4 * \left[1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \dots \right] \quad (1)$$

Write a C++ program to conduct at least 50 iterations of this infinite sequence to estimate a value for π .

- After 50 iterations, how close is the estimated value?
 - Modify your C++ program to determine how many iterations are required to get the 1st digit of π correct. (Don't manually figure this out). *Hint: you will need to think about the data types to use.*
 - How many iterations are needed to get the 1st & 2nd digit correct?
 - What about the first 3 digits?
 - Did you encounter any issues along the way, and if so why?
7. Look at the below program and answer the following questions. Then implement the code in a C++ program and check to see if your answers match what the program does. (*You can download file program from Canvas*)
- What will be the very first line of output from the program?
 - Will the program terminate?
 - If the program terminates, what will be the full output?
 - If the program does not terminate, what will the theoretical infinite output of the program?

```

1 #include <iostream>
2
3 #define EXIT_SUCCESS    0
4
5 int main(void) {
6
7     char c = 'a';
8     while (c > 0) {
9         c++;
10        std::cout << c << std::endl;
11    }
12
13    return EXIT_SUCCESS;
14 }

```

8. Look at the below program and answer the following questions. Then implement the code in a C++ program and check to see if your answers match what the program does. (*You can download file program from Canvas*)

- (a) What will the program output?
- (b) Why do you think the program produces this output?

```

1 #include <iostream>
2
3 #define EXIT_SUCCESS    0
4
5 #define LENGTH          10
6
7 void printArray(int array[], int length);
8
9 int main (void) {
10
11     int a[LENGTH] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
12     int b[LENGTH] = {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000};
13
14     printArray(a, LENGTH);
15     printArray(b, LENGTH * 2);
16
17     return EXIT_SUCCESS;
18 }
19
20 void printArray(int array[], int length) {
21     for (int i = 0; i < length; ++i) {
22         printf("array[%d] = %d\n", i, array[i]);
23     }
24     std::cout << std::endl;
25 }

```