Lab | Week 02

Overview

The Week 02 lab is for you to practice the basics of using C++. This lab contains a series of exercises, which you may have done in Java before. However, it is good to practice these same exercises to become familiar with writing and compiling C++ programs. The code snippets can be accessed from the COSC1076 Git Repository.

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.

To get the mark for this lab, show your tutor your answers up to question 7.
The challenge questions are optional.

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

```cpp
#include <iostream>
#include <string>

double add(int x, double y);

int main (void) {
    int i = 7.5;
    double d = 10.5;
    std::string s = "d + i = ";

    double sum = add(i, d);
    std::cout << s << sum << std::endl;

    return EXIT_SUCCESS;
}

double add(int x, double y) {
    return x + y;
}
```

   (a) What is the purpose of lines 1 & 2?
   (b) What value should main return?
   (c) What will the program output?

2. 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
```

3. 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)
```
     *
    ***
   *****
  ****
 ****
****
 ****
  ****
   *****
    ***
     *
```

4. 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:

(a) Read in the radius as a `float` or `double`

(b) 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 ($\pi$).

(c) Use a *function prototype* to *declare* the function.

5. In lecture we have begun writing a program for `Red7`. Recall that the game has 7 different colours, and 7 numbers in each colour.



Thus, we can number each *unqiue* card in the game from 0 to 48. Such that:

(a) Cards 0 to 6 are Red 1 to Red 7.
(b) Cards 7 to 13 are Orange 1 to Orange 7.
(c) Cards 14 to 20 are Yellow 1 to Yellow 7.
(d) Cards 21 to 27 are Green 1 to Green 7.
(e) Cards 28 to 34 are Blue 1 to Blue 7.
(f) Cards 35 to 41 are Indigo 1 to Indigo 7.
(g) Cards 42 to 48 are Violet 1 to Violet 7.
Write a C++ program that:
(a) reads in a single integer between 0 to 48 inclusive.
(b) checks the number that is read is within the correct range
(c) prints out the card (colour and number) that corresponds to this number.

6. Write a C++ program that does the following in order:
   (a) reads in 10 integers
   (b) stores the integers in an array
   (c) sums the array
   (d) prints out the result.

7. *(Challenge)* The following infinite series can be used to calculate the value of $\pi$.

$$\pi = 4 * \left[ 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \ldots \right] \tag{1}$$

   Write a C++ program to conduct at least 50 iterations of this infinite sequence to estimate a value for $\pi$.
   (a) After 50 iterations, how close is the estimated value?
   (b) Modify your C++ program to determine how many iterations are required to get the 1st digit of $\pi$ correct. (Don't manually figure this out). *Hint: you will need to think about the data types to use.*
   (c) How many iterations are needed to get the 1st & 2nd digit correct?
   (d) What about the first 3 digits?
   (e) Did you encounter any issues along the way, and if so why?
   (f) *(Extra Challenge)* Data Type Investigation. You could use a `float` or a `double` to represent the total sum. How does the use of the data type change the answers to your question?

8. *(Challenge)* 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.
   (a) What will the program output?
   (b) Why do you think the program produces this output?

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