COSC1076 | Advanced Programming Techniques

Tutorial/Lab | Week 03

Overview

The week 03 tutorial/lab is for you to practice working with:

- Functions
- Arrays
- Pointers
- References

Tutorial Questions

Look at the C++ program below, and answer the questions (on the following page).

```cpp
#include <iostream>

#define EXIT_SUCCESS    0

using std::cout;
using std::endl;

void foo(int x, double* y, char& z);

int main (void) {
   int i = 10;
   double d = 2.5;
   char c = 'e';

   int* iPtr = NULL;
   double dPtr = &d;
   cout << "iPtr = " << iPtr << ", dPtr = " << dPtr << endl;

   cout << "*iPtr = " << *iPtr << ", *dPtr = " << *dPtr << endl;
   *iPtr = 5;
   *dPtr = 4.25;
   cout << "*iPtr = " << *iPtr << ", *dPtr = " << *dPtr << endl;

   cout << "i = " << i << ", d = " << d << ", c = " << c << endl;
   foo(i, d, c);
   cout << "i = " << i << ", d = " << d << ", c = " << c << endl;
   foo(iPtr, dPtr, c);
   cout << "i = " << i << ", d = " << d << ", c = " << c << endl;

   return EXIT_SUCCESS;
}

void foo(int x, double* y, char& z) {
   x += 1;
   y += 2;
   ++c;
}
```

1. What is the purpose of lines 5 & 6?
2. The program has a number of errors.
    (a) Identify the errors.
    (b) Explain what is the error, or what might happen if the code was executed
    (c) Fix the errors.
3. What will be the output of the program, once you fix all of the errors?

Lab Questions

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.

Exercises

1. Implement the following function prototype

   ```
   bool getCharacter(char* c);
   ```

   This function reads in a single character from standard input, and places the read character into the pointer. It returns true if the character was successfully read, or false if there was a problem. (Problems might include having hit the end of input, that is, EOF).

2. Using your above function, write a C++ program that reads up to 100 characters, and puts them into an array of characters (that is, form a string). Then prints out the characters. Use a single print statement to print out the array/string.
   Make sure you properly terminate your string!

3. Modify your above program to print out the characters in reverse order.
   As an extra challenge, still only use one print statement to print out the reverse array/string.

4. Write a C++ program that does the following:
   (a) Declares, defines, and initialises an integer
   (b) Declares, defines, and initialises a pointer to the integer
   (c) Directly modifies the value of the integer
   (d) Uses the pointer to modify the value of the integer
   (e) Declares, defines, and initialises a double
   (f) Uses a function to modify the value of the double using a pointer to it
   (g) Uses a function to modify the value of the double using a reference to it

5. Using the below C++ program. *(The program can be found in the sample code for the lab)*
   (a) What will be the output of the program?
   (b) Draw a diagram to show what the program does
   (c) Run the program and see if your answer(s) are correct.

   ```
   #include <iostream>

   using std::cout;
   using std::endl;

   #define EXIT_SUCCESS    0

   int main (void) {

       int a = 7;
       int* ptr = &a;

       cout << a << endl;
       cout << ptr << endl;
       cout << *ptr << endl;

       int b = 10;
       ptr = &b;
       *ptr = 12;
       cout << b << endl;
   ```

```

    int** pptr = &ptr;
    **pptr = 20;
    cout << a << endl;
    cout << b << endl;

    *&**&pptr = &a;
    **pptr = -1;
    cout << a << endl;
    cout << b << endl;

    return EXIT_SUCCESS;
}
```

6. Look at the below function that is supposed to swap the value of two variables.

```
void swap(int* a, int* b) {
    int* tmp = a;
    a = b;
    b = tmp;
}
```

This function does not work. First, draw a diagram to show what is happening with the pointers in this function. Secondly, re-implement the function to fix it.

7. The following questions are for assignment warm-up.

   (a) Using your function in question 1, write a C++ program that:
   - Has a 2D array with 4 rows and 5 columns
   - Reads from standard input a 4x5 grid of characters and stores them in the 2D array
   - Then prints out the 2D array

   (You may assume that the grid given by input is correctly formatted)
   For example, a grid to read in might look like:

```
~~~~~
~..=~
~.=.~
~~~~~
```

   (b) Modify you above program to print out the location of every dot (`.`) character in the grid that your program reads in. That is, display that `row` and `column` of the dot. You may assume that the top-left corner of the grid is:
   - row: 0
   - column: 0

   and that rows increase downwards, and columns increase from left-to-right.

8. *(If you didn't get to this from Week 02, now is a good time to try this question).*
   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?

```
#include <iostream>

```

```
#define EXIT_SUCCESS    0

#define LENGTH          10

void printArray(int array[], int length);

int main (void) {

   int a[LENGTH] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
   int b[LENGTH] = {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000};

   printArray(a, LENGTH);
   printArray(b, LENGTH * 2);

   return EXIT_SUCCESS;
}

void printArray(int array[], int length) {
   for (int i = 0; i < length; ++i) {
      printf("array[%d] = %d\n", i, array[i]);
   }
   std::cout << std::endl;
}
```