COSC1076 | Semester 1 2021 | Advanced Programming Techniques

Lab | Week 02

Overview

The week 02 lab is for you to practice working with:

- Functions
- Arrays
- Pointers
- References

The code snippets can be accessed from the COSC1076 Git Repository.

Exercises

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.

1. Using the below C++ program.
   (a) What will be the output of the program? *(Work out the output before you run the program).* You might find it helpful to draw a diagram to show what the program does
   (b) Run the program and see if your answer(s) are correct.

```cpp
#include <iostream>

int main (void) {

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

   std::cout << a << std::endl;
   std::cout << ptr << std::endl;
   std::cout << *ptr << std::endl;

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

   b = a;
   std::cout << *ptr << std::endl;

   int* ptr2 = &a;
   int *ptr3 = &b;
   a = -3;
   b = 10;
   *ptr = 2;
   *ptr3 = 6;
   std::cout << a << std::endl;
   std::cout << b << std::endl;
   std::cout << *ptr << std::endl;
   std::cout << *ptr2 << std::endl;
   std::cout << *ptr3 << std::endl;

   int** pptr = &ptr;
   **pptr = 20;
   std::cout << a << std::endl;
```

```
    std::cout << b << std::endl;
    std::cout << *ptr << std::endl;
    std::cout << *ptr2 << std::endl;
    std::cout << *ptr3 << std::endl;

     // Challenge
    *&**&pptr = &a;
    **pptr = -1;
    std::cout << a << std::endl;
    std::cout << b << std::endl;

    return EXIT_SUCCESS;
}
```

2. The following program contains a number of errors. Explain what errors are, and fix them.

```
#include <iostream>

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;
}
```

3. Write a C++ program that does the following:
   (a) Declares and initialises an integer
   (b) Declares 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 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

4. In this question you will develop a small program for reading characters from standard input.
   (a) Write a simple program to read in a single character from standard input and print it to standard output.

   (b) Modify your program to instead use the following function to do the reading from standard input:

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

   This function reads in a single character, and places the read character into the given pointer. It returns true if the character was successfully read, or false if no character was read, due to having hit the end-of-input, that is, EOF (`^D`).

   (c) Modify your program to reads up to 100 characters, and put the characters into a character array:

   ```
   char characters[101];
   ```

   Then use a *single* output statement to print out the array/string, without using a loop.
   Make sure you properly terminate your character string!
   As an extra challenge, still only use one print statement to print out the reverse array/string.

5. Implement a C++ Class to represent a Card in Red7. Recall that a card has three components: For our implementation, we want to have 4 methods that return:
   - The card's number
   - The card's colour, represented as a number (using the codes for colours from lectures)
   - The card's colour as a string, ie `red`, `orange`, etc.
   - The rule corresponding to the card's colour (see image)

   You will also need to specify a suitable constructor for the class, and decide on appropriate fields for the class. The code repository has a starting point for this class.
   To test your card class is correct, create represent the card `Red 7`, and print out its number, colour (as both an integer and string), and rule.

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. *(Challenge)* Modify your program from question 3 to:
   (a) Reverse the string of characters *in-place*. That is you cannot use another array.
   (b) Print out the characters both the original and reverse order.
   (c) Make the arrays of characters be as small as possible. That is, if there are fewer than 100 characters entered by standard input, the array is smaller than 100 character long. It should still be printed out in both the original and reverse order.