COSC1076 | Advanced Programming Techniques

Tutorial/Lab | Week 04

Overview

The week 04 tutorial/lab is for you to practice working with:

- Typedef
- Array representation in C/C++
- Pointers
- References
- Allocating and De-allocating memory

Tutorial Questions

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

Example.h

```
typedef double& DblRef;

class Example {
public:
    Example(double value);

    void setValue(DblRef value);
    double getValue();

private:
    double* ptrValue;
};
```

Example.cpp

```
#include "???"

Example::Example(double value) {
    this->ptrValue = new double(value);
}

void Example::setValue(DblRef value) {
    this->ptrValue = value;
}

double Example::getValue() {
    return this->ptrValue;
}
```

init.h

```
typedef int* IArray;

void initialise(IArray array, int length);
```

init.cpp

```
#include "???"

void initialise(IArray array, int length) {
   for (int i = 0; i < length; ++i) {
      array[i] = 0;
   }
}
```

main.cpp

```
#include <iostream>
#include "???"

#define EXIT_SUCCESS    0
#define LENGTH          10

int main (void) {
   Example* example = new Example(7.5);

   double dbl = 10;
   example->setValue(dbl);
   std::cout << example->getValue() << std::endl;

   IArray intArray;
   initialise(intArray, LENGTH);

   return EXIT_SUCCESS;
}
```

1. How can you compile all of these files into a single C++ program?
2. Complete the `#include` statements at the top of each code file.
3. Explain what happens on line 8 of `main.cpp`. Draw a diagram to show all memory that is created.
4. What is an alternative way for writing line 11 in `main.cpp`?
5. What is this `Example` class missing? Explain why this is a serious issue.
6. What is missing in `main.cpp`?
7. The files contain further errors. Find and fix the errors.
8. What will be the final output of the program?

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. Write a C++ program that, for each of the following types:
   - `int`
   - `char`
   - `std::string`
   - Array of `double`'s
   - Array of *pointers* to `float`'s

   does the following:
   (a) Allocated memory on the *Program Call Stack*
   (b) Initialises their values, to sensible values (your choice)
   (c) Prints out the contents of variables
   (d) Ensures all memory is correctly de-allocated at the end of the program.

2. Using the same types listed in Question 1 above, write a C++ program that:
   (a) Creates memory on the *Heap* for each variable. (Remeber to use pointers as necessary).
   (b) Initialises their values, to sensible values (your choice)
   (c) Prints out the contents of variables
   (d) Ensures all memory is correctly de-allocated (deleted/freed) at the end of the program.

3. What are the differences you had to use between your programs in Questions 1 & 2?

4. Write a C++ Class to represent a single Video Game character. The character has two attributes, a name and a health-point total.
   You should:
   (a) Use a header file for the class declaration
   (b) Use a code file to implement the methods of the class
   (c) Use appropriate constructor's, deconstructor's, fields and methods to define the class.
   (d) You should be able to change the characters health-points
   (e) The name of the character must not be able to be changed

5. Using your class in Question 4, write a C++ program that:
   (a) Allocated memory for a video game character object on the *Program Call Stack*
   (b) Initialises the character with 50 health-points.
   (c) Changes the health-points of the character to 30.
   (d) Prints out the name and current health-points of the character.
   (e) Ensures all memory is correctly de-allocated at the end of the program.

6. Using your class in Question 4, write a C++ program that:
   (a) Creates an object for a video game character on the *Heap*
   (b) Initialises the character with 40 health-points.
   (c) Changes the health-points of the character to 60.
   (d) Prints out the name and current health-points of the character.
   (e) Ensures all memory is correctly de-allocated (deleted/freed) at the end of the program.

(Question continue on the next page)

7. *(This question will help with Assignment 1)* Extends the class you created in question 4 to allow the video game character to maintain an inventory of items. The inventory will should be represented as a concrete array of `std::string`'s (shown below), that is, the length of the array is specified in the class.

```
Class VideoCharacter {
    // ... your existing implementation

private:
    // ... your existing private methods and fields

    std::string inventory[LENGTH];
}
```

The string is the name of each item in the inventory. For the purpose of this question, assume a character can hold no more than 10 items.
Complete the following tasks:

(a) Change your definition in Question 4 to allow:
   i. Items to be added to the inventory
   ii. Return the number of items in the inventory
   iii. Get the i'th item in the inventory

(b) Update your implementations in Questions 5 & 6. Make sure you also update the constructor's and deconstructor's.

(c) Did you need to "new" (create) memory for the array? Why?