

CSCE 121 Assignment #4

due October 4th to eCampus

Objectives: This assignment gives you some experience with writing C++ programs that use fundamental computation concepts: iterations, comparisons, functions, and vectors. Also, you will have an opportunity to write code that catches errors and throws exceptions.

All work should be pre-graded in the labs and turned in to eCampus by the deadline. For programs, you need to turn in only the source code (not object or executable files). Your code will be tested using a g++ compiler. Therefore, you can develop your program in Visual Studio or Xcode, but make sure your code also compiles using g++ and runs on a Linux machine.

1. (5 points) Create a text file, called README, in which you provide:

- Your first name, last name, UIN, section number, user name, e-mail address
- State the Aggie Honor statement:

I certify that I have listed all the sources that I used to develop the solutions and code to the submitted work.

On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.

Your Name

Date

- List any resources used such as webpages titles and URLs. Do not mention the textbook and discussions with the Instructor, TA, or Peer Teachers. Describe in details the type of help you have obtained from each resources.
- List any known problems with the assignment you are turning in. For example, if you know your code does not run correctly, state that. This should be a short explanation.
- Write how you tested your programs for correctness and used exceptions for the error handling.
- Submit to eCampus an electronic version of the README file along with C++ code for the problems listed below and a hard copy to your TA.

2. (Lab 30 points) Complete the drills 1, 2, 3, pp. 297 and 298.

The C++ code for the above problems should be completed during the lab time, and presented to your TA or PTs before submitting to eCampus.

3. Write a program using a single file containing three namespaces **X**, **Y**, and **Z** so that the following correctly:

[Click here to view code image](#)

```
int main()
{
    X::var = 7;
    X::print();           // print X's var
    using namespace Y;
    var = 9;
    print();              // print Y's var
    {
        using Z::var;
        using Z::print;
        var = 11;
        print();          // print Z's var
    }
    print();              // print Y's var
    X::print();           // print X's var
}
```

Each namespace needs to define a variable called **var** and a function called **print()** that outputs to **cout**.

2. Write three functions `swap_v(int,int)`, `swap_r(int&,int&)`, and `swap_cr(const int&,const int&)` have the body

```
{ int temp; temp = a; a=b; b=temp; }
```

where **a** and **b** are the names of the arguments.

Try calling each swap like this

[Click here to view code image](#)

```
int x = 7;
int y = 9;
swap_?(x,y);           // replace ? by v, r, or cr
swap_?(7,9);
const int cx = 7;
const int cy = 9;
swap_?(cx,cy);
swap_?(7,7,9,9);
double dx = 7.7;
double dy = 9.9;
swap_?(dx,dy);
```

```
swap_?(7.7,9.9);
```

Which functions and calls compiled, and why? After each swap that compiled, print the call to see if they were actually swapped. If you are surprised by a result, consult §8.6.

Important: A file name should correspond to the exercise number, e.g. LastName_UIN_D1-p297.cpp where LastName is your last name and UIN is your UIN number.

1. (35 points) Write a program that computes the sequence of Fibonacci numbers. The formula for generating the next Fibonacci number is: $F_n = F_{n-1} + F_{n-2}$, where $F_1 = 1$ and $F_2 = 2$. For example, $F_3 = F_2 + F_1 = 2 + 1 = 3$. You will notice that at some point Fibonacci numbers are too large and they do not fit in type **int**. This is called the integer overflow. When they become negative (or non-increasing), stop the program. Please use three different functions, one for each type.
- How many Fibonacci numbers fit in type **int**? What is the value of the largest one?
 - How many Fibonacci numbers fit in type **long**? What is the value of the largest one?
 - How many Fibonacci numbers fit in type **long long**? What is the value of the largest one?

Interesting note: “13 year old makes solar power breakthrough by harnessing the Fibonacci’s sequence”, see

<http://inhabitat.com/13-year-old-makes-solar-power-breakthrough-by-harnessing-the-fibonacci-sequence/>

2. (20 points) Write a recursive function `decimal_to_octal(int n)`. This function should take a non-negative integer and return its octal form. Test the function in `main` using integers. Use the prefix `0` when you display a number in the octal form. Write in the README file about what

you have learned by writing this assignment and what results of your testing are. Write also about your (possible) problems, errors or mistakes.

3. (40 points) Reverse the order of elements in a vector of integers using two functions:

- (a) The first function takes the input vector `a` and return a vector `b` with the input data but in the reverse order, e.g. input: 1, 2, 3 and output 3, 2, 1.
- (b) The second function is a *recursive* one and it reverses the input data in place (without using an additional vector).

Test the first and second algorithm using the same vector and compare results.

Important: A file name should correspond to the assignment number and exercise number, e.g. LastName_UIN_A4-Ex5.cpp.

Submit all the files together as a zip/tar file to eCampus for grading.