CS 240 Homework 1 **Fall 2017**

All answers must submitted electronically into Blackboard, in a .tar.gz file named according to the CA specifications (replace "CA" with "HW"). Please put your answers in separate files, as described below, and include a makefile that builds all of your code.

Recursion and Templates [40 pts]: These questions contain some programming! The only way to learn recursion is to do it. Don't wait until the last minute, please.

(1) [10 pts] Write a C++ program that contains, and calls from `main()`, a recursive function to print the first N multiples of 3, in order. Include this entire program (no header or other files) in a file named `HW1-1.cpp`.

(2) [10 pts] Write a C++ program that contains, and calls from main(), a recursive function to print the first N multiples of 3, in reverse order. Include this program (no header or other files) in a file named `HW1-2.cpp`.

(3) [5 pts] Do some googling to find out what tail recursion is. *In your own words*, what is tail recursion and why is it important? (Submit this answer by itself in a text file named `HW1-3.txt`.

(4) [15 pts] Write a very simple C++ class called `Team`, which contains a `string` and two `unsigned integers`, one for wins and one for losses. Create a C++ vector of `Team` objects. Next, write your own templated C++ non-member function that sorts items of type T, contained in a C++ `vector`, using your own implementation of merge sort. Any access to vector items must be done through an `iterator`. Overload one or more of the `Team` comparison operators (e.g. `operator<()` (less than) to compare by winning percentage (wins / (wins + losses)). Show that your code uses C++ templates properly, by also instantiating a vector of ints, and sorting those with the same merge sort function. Submit this entire program in a file named `HW1-4.cpp`.

Trees [40 pts]

(5) [15 pts] Draw a 12-node unbalanced binary search tree containing 12 different integers in the range 1-99. Please include your picture in a PDF file named `HW1-5.pdf`. (If you really do not want to "draw" the program using a tool such as Powerpoint or Word, you may do so by hand, take a picture of it, and convert using jpg2pdf (or something). But your picture must be submitted in a pdf file.)

(6) [15 pts] Draw a 9-node balanced, but not complete, binary search tree containing 9 different strings. Please include your picture in a PDF file named `HW1-6.pdf`. (2 bonus points if the inorder traversal of your tree produces an intelligible 9 word sentence!)

(7) [10 pts] (a) Draw a *perfect* 15-node binary search tree containing the first 15 positive integer multiples of 10.
(b) Show the result of inserting the numbers 105 and 115, and then removing 40 and 100. You may draw multiple trees, to be safe (for partial credit), or you may draw just the result of the 4 operations. Please include your picture in a PDF file named `HW1-7.pdf`.

Operation Counting and Runtime Complexity [20 pts]

(8) [20 pts (5 pts each)] Calculate the frequency count and corresponding Big Oh runtime complexity of the following code snippets. N is the problem size, not a constant, so your answers should be written in terms of N. Type your answer, with explanation, into a file named `HW1-8.txt`.

(a)

```
int i, j, k;
for (i=0; i < N; i++)
   for (j=0; j < N; j++)
      for (k=0; k < N; k++)
         cout << i << j << k << endl;
```

(b)

```
int i, j, k;
for (i=0; i < N; i++)
   for (j=0; j < N; j *=2)
      for (k=0; k < N; k++) {
         cout << i << j << k << endl;
```

(c)

```
int i, j, k;
for (i=0; i < N; i *= 2)
   for (j=0; j < 100; j++)
      for (k=0; k < N; k++) {
         cout << i << j << k << endl;
         k = k * 2;
      }
```

(d)

```
int i, j, k;
for (i=0; i < 100; i *= 2)
   for (j=0; j < N; j *= 2)
      for (k=0; k < N; k *= 2) {
         cout << i << j << k << endl;
      }
```