

## CISC181H Spring 2009 Lab07

- Write a program for each of the following problems. Be sure to save every separate program. All programs must be properly commented and indented (see Assignment Standards on the class website). Ask your TA for guidance.
- Your TA is an excellent resource. You can also ask questions about lab problems in lecture and in **office hours**. Students who attend Matt's hours this week get 2 points added to their last lab grade, plus free answers to questions.

### Programs

1. On Monday we saw that classes with pointers to dynamically allocated memory can result in aliasing, which is dangerous when it falls into the wrong hands.

Read the file `constructors3.cc` and predict the output of each line in `main()` before you compile and run. Then go over your predictions with another student *before* you compile and run. Write your predictions on paper (by hand) and **hand them to your TA before 9:20 a.m.** This task should occupy 5-10 minutes of your lab time.

Now compile and run.

Based on your observations, design ways to avoid the aliasing that occurs in this program.

2. Copy your linked list code into a new directory (you can only have one makefile per directory). Modify the Nodes to hold Cows (with name and weight) instead of numbers, and implement the Big Three for Cows. Modify test code appropriately and demonstrate.
3. Write a comparator method for Cows. It should return true iff Cows have the same name and weight. It takes a single Cow reference as a parameter.
4. Write a recursive function that will separate and indent matched sets of braces. For example, the input C-string `{{}} {{{}} {{{}}}` will produce:

```
{
  {
  }
  {
    {
      {
      }
    }
  }
}
```

I suggest using the degree of indent as a parameter.

5. Write a function that raises some numeric base to a positive integer exponent. Your function may be recursive or iterative<sup>1</sup>. Now make your function run in  $O(\log n)$  time by dividing the problem in two whenever possible. This may require thinking - what can you divide, and when? I suggest you start with a pencil and paper and smallish numbers. I also suggest a separate `square(x)` function as a helper.
6. Copy `constructors4.cc`. Figure out which line in `main` causes a problem and comment it out so the program compiles and runs correctly.
7. Copy `constructors5.cc`. Write a class `B` that contains a public data member of type `A1`. Declare a `B` and print the value of its `A1`. Compile and run.

Now declare a class `C` that contains a public data member of type `A2`. Declare a `C` and print the value of its `A1`. Describe what happens. Then read chapter 7.1 on initializers and change only the constructors of `C` and `A2` so that it works.

You should have a total of 7 programs named `lab07.1.cc` to `lab07.7.cc`, plus any makefiles and written answers specified above. Make a single script file (see `lab00` for the instructions) where you `cat`, `compile`, and run lab code in its final form.

Submit all 7 program files *and* your script on WebCT by midnight before your next lab. Give the paper version of the complete script file **only** on paper to your TA at the **beginning** of your next lab. Note: `Cat`, `compile`, and run each program in order - do *not* `cat` all programs, then `compile`, etc.

---

<sup>1</sup>but nobody ever says “cool” when you write a for loop.