

## CS2260, Fall 2005, Final

Time 90 min. All questions are weighted the same. Use extra paper as needed, but make sure to identify each answer.

**YOU MUST RETURN THIS PAGE. NAME** \_\_\_\_\_

- 1 Implement class for boxes, where each box has private length and width (integers) as well as a label (label must be array of char or string but allocated on the heap). Make sure the following would work:

**Box b; // 0x0 "Dummy Box"**

**Box b("BDay Cake",2); // 2x2 "BDay Cake"**

**Box b("Surprise",1,3); //1x3 "Surprise"**

- 2 Implement copy and converter from `char[]` so that  
**Box b="Shoebox"; // would create 1x1 "Shoebox"**
- 3 Add capability to ask for the number of box objects:  
**cout << Box::howManyBoxes(); // prints # objects**
- 4 Overload binary minus so that **b1-b2** will return a
  - if **b2** would fit inside of **b1** (in either direction), **b1-b2** would just return a copy of **b1**
  - otherwise **b1-b2** would create and return (by copy) the smallest box that could fit **b2** inside, labeled "**BoxWrapper**".
- 5 Overload `<<` and `>>` so that input will prompt and read new values for all 3 attributes, and output will display values of all 3 attributes.
- 6 For the Box above, if anything else needs to be implemented (to avoid wrong box counters and memory leaks) do so now.

For **CBox** below us public enumeration for color.

- 7 Create **CBox** from **Box**. **CBox** is colored box, and it has color (use `char[]` w/o dynamic allocation). Make sure the following would work:

**CBox cb; // white 0x0 "Plain Box"**

**CBox cb("Red",2,3,"My Red Box"); // Red 2x3 "My Red Box"**

- 8 Overload binary `+` so that  
**CBox cb("Red",2,3,"My Red Box");**  
**cb+"Blue"; // cb is now blue**
- 9 Assume top level functions  
**void tryBox(Box &);**  
**void tryCBox(CBox &);**  
and two objects  
**Box b;**  
**CBox cb;**  
Which object can be send to which function?

- 10 Provide a class template for a **Pair** of two objects of the same type. Provide a constructor with two arguments, making such a pair. Provide a method **bool isFirstBigger()** which returns true if the first element is bigger than the second.  
**Pair<int> p(3,2);**  
**if (p.isFirstBigger())**  
    **cout << "first element bigger";**
- 11 What must be true for the above to work for pairs of any type?
- 12 Now provide a template for a pair of two different elements, everything else the same as in #10.
- 13 Suppose you have class **Car** with << and >> overloaded. Write an application program to read 3 cars, put them into a container. Then the program reads another car, and prints a message whether the car is in the container or not.
- 14 Write a short program that reads all integer numbers from a file up to EOF. The program asks the filename first. The assumption is that there is no more than 10 numbers. The program reads into an array, then prints them back in reverse.
- 15 In the above, instead of printing back in reverse, do the following: the array is sorted and printed out using STL (yes use only STL for this).