

CS274, Winter 2001, Test 1

Time 70 min. 5 Questions, 20 pc each. Use extra paper as needed, but make sure to identify each answer.

YOU MUST RETURN THIS PAGE. NAME _____

- 1 The user types some floating numbers on the keyboard and then simulates the end of a file. Write a complete program to determine and display the smallest number entered.
- 2 There is a text data file of the following format (no spaces in Names and numbers are integers):
Name1 number1
Name2 number2
etc.
The program asks and reads the name of the data file from the keyboard and then produces the following standard output:
Name1 number1 pct1%
Name2 number2 pct2%
where pct% is the % of how much number1 is of the total of all the numbers, Names are left justified, the other data is right justified, and all fields are 20 spaces. Write a complete program w/o functions.
- 3 Implement a class for a `Building`. Each building has a name (use dynamically allocated array of `char`) and `numberOfRooms`. They are private. Have a single constructor with all default arguments (defaults are "NoName" and 0). Show the header and the source files separately.
- 4 For the above, add overloaded `<<` and `>>` operators. Show all changes only.
- 5 Derive `House` from `Building`. `House` also has `owner` - use string. Have a single constructor with all default values (defaults are "Some Name" for building name, "John" for owner, 3 rooms). Show the header and the source file.
- 6 What would be needed to make `Building` an abstract class?
- 7 Suppose we want polymorphic `show()` method for both `Building` and `House` applicable for dynamic binding. The methods just displays all the information about the `Building/House`. Show all changes needed (show only changes in every file) and the implementations.
- 8 Implement a copy constructor for `Building`.
- 9 Implement a copy constructor for `House`.
- 10 Implement a function that will allocate a `House` or a `Building` based on an integer argument (0 for `House`, anything else for `Building`). The function will allocate the

object, and then return it to the calling program. Show the original call. Do this:

- a) by reference
- b) by pointer

- 11 Use the `vector` container to read 20 numbers from the keyboard, `sort` them, and the print them back. Implement the complete program.
- 12 Implement a template class for a pair of elements of the same type. Provide `match()` method - will return `true` if both elements are the same.
- 13 Extend the above with `<` and `+` binary operators. `<` will evaluate to `true` if the smaller element from the first pair is smaller than the smaller element of the second pair. `+` will create a new pair with the first element being the sum of the first elements, and the same for the second element.