

Name: (as it would appear on official course roster)		
Umail address:	@umail.ucsb.edu	section 9am, 10am, 11am, 12pm
Optional: name you wish to be called if different from name above.		
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")		

h04: Class Design

ready?	assigned	due	points
true	Tue 10/08 02:00PM	Tue 10/15 02:00PM	50

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

MAY ONLY BE TURNED IN IN THE LECTURE LISTED ABOVE AS THE DUE DATE,
OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments;
in place of that, we drop the lowest scores (if you have zeros, those are the lowest scores.)

Reading: Class Design, DS 2 In CMPSC 24, you likely already read parts of Chapter 2. I want to invite you to read it again, for two reasons: (1) to review parts you may have skipped or skimmed over the first time (2) to revisit things that you may have read many weeks or months ago, and that may not have fully made sense to you at the time.

I've included some questions here on a few of the important points from Chapter 2, but these are NOT the only things you need from Chapter 2. I don't have enough room on the page to give you homework over everything from Chapter 2 you need to learn. Read few this material a few times between now and the first exam.

1. (10 pts) Fill in the information in the header. The following are required to get the 10 "participation" points.
 - o Filling in your name and umail address.

Also: For paper submission PLEASE submit on ONE SHEET OF PAPER, double-sided if at all possible. If you must submit on two printed sheets write name on BOTH sheets and no staples, paperclips, or folded corners.

2. According to Chapter 2 of DS:
 - a. (4 pts) In C++ if you write a class, but do not specify any constructors, what happens?
 - b. (4 pts) When you implement a so-called "default constructor" in C++, what *specific* capability does this give the user of your class? (By specific, I mean what capability does the *default constructor* give that is distinguished from other kinds of constructors?)
 - c. (4 pts) Suppose you are writing a class called Student that has two private members, 'std::string name' and 'int perm'. Write the function prototype (function declaration) ONLY for a default constructor for Student. For full credit, include the semicolon that terminates the prototype, but DO NOT write the body of the constructor

1
h04
CS32 F19

2

h04
CS32 F19

3. Before enrolling in CMPSC 32, you should have learned about stacks in CMPSC 24 (or an equivalent course). I'm going to tell you a few things about an example stack implementation as a C++ class, then ask a few questions—these are designed to check your understanding of some important ideas about ADTs and C++ classes from Chapter 2 of DS.

Suppose we implement a class for a stack of integers, called `IntStack`. Our implementation uses a variable `int stack[MAX_SIZE];` and a variable `int size;`. The constant `MAX_SIZE` may be declared as `const int MAX_SIZE=100;`, for example. I will have methods `push`, `pop`, `isEmpty` and since my implementation has a max size, `isFull`.

If I declare a method with the function prototype `bool isEmpty() const;`, I have the option of putting the function body either in a separate `IntStack.cpp` file, or directly inside the `IntStack.h` file like this:

```
bool isEmpty() const { return size==0; }
```

- a. (2 pts) According to C++ recommended practice, in which section of the class should we find the declarations of `int stack[MAX_SIZE];` and `int size;`?

Circle one: `public` `private`

- b. (2 pts) When we put the method body (e.g. `{return size==0;}`) for `isEmpty` inside the class declaration (i.e. in the `IntStack.h` file) that makes `isEmpty` a special type of member function (three word phrase with initials IMF). What does IMF stand for?

- c. (3 pts) What is one thing that is different about IMF from the perspective of how they are treated by the compiler?

- d. (4 pts) What is the significance of the keyword `const` that appears after `isEmpty`?

- e. (4 pts) Would `const` be appropriate for the `pop` operation?

Circle one: `yes` `no`

Briefly explain your answer.

- f. (4 pts) Would `const` be appropriate for the `isFull` operation?

Circle one: `yes` `no`

Briefly explain your answer.

4. DS Chapter 2 refers to something called a macro guard. This is a set of two pre-processor directives that appear at the start of a file, and one that appears at the end of a file. DS's macro guard for a file `throttle.h` uses a symbol `MAIN_SAVITCH_THROTTLE`, but it would be more traditional to use a symbol such as `_THROTTLE_H_` (the name of the file, with underscores before, after, and replacing the dot.)

- a. (4 pts) Write a macro guard for a file called `IntStack.h`, using the symbol `_INTSTACK_H_`. Put an ellipsis (...) in the place between the first two lines and the last line.

- b. (5 pts) What is the purpose of a macro guard?