First name (color-in initial)	A	B	C	D	E	F		G	H	I	J	K	L		M	N	O	P		Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1 or 2)	first name initial	last name initial
Last name (color-in initial)	A	B	C	D	E	F		G	H	I	J	K	L		M	N	O	P		Q	R	S	T	U	V	W	X	Y	Z			

H03: Due Wednesday, 10.15 in Lecture

Functions (Ch 4.1-4.3)
Assigned: Mon 10.13   Total Points: 50

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE,
or offered in person, for in person grading, during instructor or TAs office hours.
See the course syllabus at https://foo.cs.ucsb.edu/16wiki/index.php/F14:Syllabus for more details.

---

(1) (10 pts) Fill in the information below. Also, fill in the A-Z header by

- **coloring in** the first letter of your first and last name (as it appears in Gauchospace),
- writing **either 9,10,11,12,1 or 2** to indicate your **discussion section (lab)** meeting time
- writing your **first and last initial** in large capital letters.

All of this helps us to manage the avalanche of paper that results from the daily homework.

name:	
umail address:	@umail.ucsb.edu

If you collaborated with AT MOST one other person on this homework, write his/her name below. She/he should also have your name on his/her paper.

Reading: *Read Chapter 4, Sections 4.1 through 4.3. (If you don't have a copy of the textbook yet, there is one on reserve at the library.)*

Then, answer the following questions. Be sure to check both sides.

2. (5 pts) What line of code (hint: it's a pre-processor directive) do you include at or near the top of a C++ source file if you want to calculate a square root in your program?

3. (4 pts) Fill in the blank:

   According to Savitch, a "function invocation" is a fancy of way of saying function _____.

4. (4 pts) Fill in the blank:

   According to Savitch, the input to a function,
   rather than coming from `cin`, is through its _____ _____.(two words)

5. There are two ways to do type casting in C++—one that uses the `static_cast<some_type>` notation, and another that uses the `(some_type)` notation, where `some_type` is actually something such as `int`, `double`, etc.. Though the book correctly notes that you should only use the `static_cast` syntax in C++ programs, when writing C, you may need the other syntax. (That older syntax is still used in C; the newer C++ `static_cast` syntax is typically NOT available for use in C)

   a. (5 pts) Briefly, in your own words, what does type casting actually *mean*, i.e what is it *for*, or what does it *do*?

   b. (5 pts) Suppose you have a variable declared as `int count;` and another variable declared as `int sum;`. Assume that sum and count have both been given values, and that you've already checked, count is not zero.

      Write a line of code that declares a variable `avg`, of type `double` and assigns it to `sum` divided by `count`, but use a static_cast to convert both variables to values of type `double` before the division takes place.

   c. (5 pts) Now write the same line of code, but this time use the older C++ style of type casting (the one that you may have just learned is typically also used in C, even now.)

6. Savitch discusses three concepts that are very important to keep straight, and not confuse: (a) function **declaration** (also called function **prototype**), (b) function definition (c) function **call**. Here is a short C++ program, with line numbers. Please indicate after the program which line number (or range of line numbers, e.g. 3-5 or 7-14) contains the function prototype, function definition, and function call for the `isDivisibleBy` function. (Note: The program is also currently online at: https://dbgr.cc/t3 if you want to run it in a web browser.)

```
1.    #include <iostream>
2.    using namespace std;
3.
4.    bool isDivisibleBy(int a, int b);
5.
6.    int main() {
7.       cout << "result for (15,5) is " << isDivisibleBy(5,15) << endl;
8.       cout << "result for (15,5) is " << isDivisibleBy(5,15) << endl;
9.    }
10.
11.   bool isDivisibleBy(int a, int b) {
12.      return ( a % b == 0 );
13.   }
```

   a. (4 pts) line number(s) or line number range of function prototype (also called "function declaration") for `isDivisible`

   b. (4 pts) line number(s) or line number range of function definition for `isDivisible`

   c. (4 pts) line number(s) or line number range of function calls for `isDivisible`