

Name: (as it would appear on official course roster)		
Umail address:	@umail.ucsb.edu	section
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: Chapter 4: Predefined and programmer defined functions

ready?	assigned	due	points
true	2019-19-10 9:00	Thu 10/17 12:30PM	30

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

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE, OR IF APPLICABLE, SUBMITTED ON GAUCHOSPACE. There is NO MAKEUP for missed assignments;

Read Chapter 4, sections 4.1 - 4.3. Please turn in your homework in lecture.

PLEASE MARK YOUR HOMEWORK CLEARLY, REGARDLESS OF IF YOU WRITE IT OUT IN INK OR PENCIL!

- (4 pts) What is type casting and how is it performed in C++?
- (4 pts) Which of these uses of type casting will NOT ensure that f is 1.5? Answer should be (ex1), (ex2), (ex3), or (ex4) (or a combination of those).

```
int a(1), b(2), c(2), d(2), e(2);
double f;

f = (a + b)*c / static_cast<double>(d + e); // (ex1)
f = static_cast<double>(a + b)*c / (d + e); // (ex2)
f = (a + b)*static_cast<double>(c) / (d + e); // (ex3)
f = static_cast<double>((a + b)*(c) / (d + e)); // (ex4)
```

Please:

- No Staples.
- No Paperclips.
- No folded down corners.

1
h04
CS16 F19

3. (6 pts) We talked about three concepts that are very important to keep straight, and not confuse: (a) function declaration, (b) function definition, and (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.

```

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      return 0;
10 }
11
12 bool isDivisibleBy(int a, int b) {
13     return ( a % b == 0 );
14 }

```

2

h04

CS16 F19

4. (6 pts) The infinite series: $s = 1 + (2/3) + (4/9) + (8/27) + \dots$ is a geometric series that converges to a whole rational number (i.e. like 2 or 3 or 4). Below is an unfinished C++ program that will calculate s to the k th position (so, for example, if $k = 1$, then $s = 1 + (2/3) = 1.666\dots$). Fill in the missing code:

```

#include <iostream>
#include <cmath>
using namespace std;

int main(){
    double s(0.0);
    int k(0);
    cout << "Enter k: ";
    cin >> k;
    for ( _____ ) {

        s = _____; }

    cout << "Series converges to: " << s << endl;
    return 0;
}

```

5.(10 pts) Write a function declaration and a function definition for a function named 'check_descending' that takes three arguments of type `int` and returns `true` if the arguments are in descending order; otherwise, it returns `false`. You can use any type of C++ statements that we've gone over in class so far to accomplish this goal.