

Name: \_\_\_\_\_ Section: \_\_\_\_\_ Grade: \_\_\_\_\_

Answer these questions while viewing the assigned videos. **Not sure of an answer?** Ask your instructor to explain **at the beginning of the next class session**. You can then fill in your answer, still for full credit. (But no fair doing that unless you attempted the question first.)

**Videos for this quiz:** [www.rose-hulman.edu/class/csse/binaries/C-Videos/session2/](http://www.rose-hulman.edu/class/csse/binaries/C-Videos/session2/)

**Video 1: First C Program (and for loops)** (10 minutes, 52 seconds)

1. What symbols do you use for a multi-line comment in C? (Give a short example that uses them.)
2. What symbols do you use for a single-line comment in C? (Give a short example that uses them.)
3. A **#include** statement in C is like a \_\_\_\_\_ statement in Python.
4. Every function in C has a return type. **True** or **False?** (Circle your choice.)
5. Write a function called **foo1** that *prints* the number **78**. (Note: What will the return type of this function be?)
6. Write a function called **foo2** that *returns* the number **78**. (Note: What will the return type of this function be?)
7. Write a function called **foo3** that has two parameters, both floating point numbers, and returns the product of those two numbers.

8. A **for** statement in C has 3 parts. **True** or **False?** (Circle your choice.)
9. The three parts of a **for** statement are separated by the \_\_\_\_\_ character. (Hint: NOT a comma!)
10. When does the first part of a **for** statement execute? (Before the loop, beginning of loop, middle of the loop, end of the loop, after the loop, or some other time?)
11. When does the third part of a **for** statement execute? (Before the loop, beginning of loop, middle of the loop, end of the loop, after the loop, or some other time?)
12. The following statements accomplish the same thing: **True** or **False?** (Circle your choice.)
- ++k**  
**k++**  
**k = k + 1**
13. Write a function called **foo4** that has two parameters, the first a floating point numbers and the second an integer, and returns the first number raised to the second number.  
For example, **foo4(2, 5)** should return **32**. (Use a **for** loop to solve this problem.)
14. Review the FirstExample code from your Session24 project. Write here any questions you have about any part of it.