# CS 172 Review for Exam 1 (Spring 2016)[[1]](#footnote-1)

Exam 1 is a hands-on practical test. You are allowed to look at any previous programming solution for the exam. Also, if you are using the **string** class, you are allowed to look at <http://www.cplusplus.com/reference/string/string/>. The exam will cover **chapters 9 and 10**, but it will assume you are competent with all concepts covered in CS-171. **Please read the textbook** and **review the concepts** described in this document.

**C++ Review (all concepts covered in CS-171)**

* Know how to write **for** and **while** loops, and **if** statements.
* Know how to define functions with parameters.
* Know how to define and use arrays
* Know how to specify mathematical and boolean expressions with the standard C++ operators, use the pseudo-random number generator, etc.

**Designing Software with Objects**

* Controlling complexity on large projects is a challenge. Know how Object Oriented design help us control complexity?
* How do you identify objects, properties and behaviors from a software requirements specification?
* What are the class access control keywords? E.g. **private** and **public**. What are they used for?
  + Know that there is a third keyword, **protected**, but you will not be tested on this.
* How do you define a member variable (i.e. property) in C++? How do you define a member function (i.e. behavior) in C++?
* A class is blueprint (or template) for making .
* What is the purpose of the **‘.’** operator?
* What is the purpose of a class **constructor** and how do you **define one**? Know how to define **overloaded constructors**.
* Know what is a header **inclusion guard** and how to define them
* Know how to put a class declaration in a **class definition header file** (\*.h file), and put method implementations in a **class implementation file** (\*.cpp file)
* Know how to read and interpret **UML** class diagrams.

**The string Class – An Example of a Good O.O. Design**

* Know how to use the methods available in the string class:
  + You will be allowed to refer to <http://www.cplusplus.com/reference/string/string/> if you need too.

**Passing Objects to Functions, Arrays of Objects, Modeling Object Composition Relationships**

* What is the difference between function parameters that pass an object “**by value**” and those that pass an object “**by reference**”?
* What is the purpose of **const**? Why is it useful and where can it be used?
* What are **static** member variables/methods?
* Know how to compose 2 classes together. For example, when the software requirements say that a **Student** *has-a* **Faculty** advisor, this will result in a design where a **Student** class contains a member variable of a **Faculty** class type. Know how to read and interpret **composition (HAS-A) relationships** in **UML** diagrams.
* How do you define an **array of objects**? How do you **access** individual object elements in the array? How do you **initialize** an array of objects?

## Practice Exercise for Exam 1[[2]](#footnote-2)

Log in to GitHub and open up my public repository at <https://github.com/MTBellAtWhitworth/Dice>. There’s a whole VS2015 project there + UML diagram (in Adobe PDF form) showing how to write a simple dice roller. Examine the UML. Now, open up the VS project and take a look at Dice.h, Dice.cpp, and main.cpp. Understand the code. Try to break it, then fix it again. Ask yourself how you’d write something similar. Ask yourself how you’d use this little program to write a simple dice game.

1. Barely modified from Tucker, Exam Review for Summer 2015. [↑](#footnote-ref-1)
2. REALLY make sure you can do this --- it’s excellent preparation for the exam! [↑](#footnote-ref-2)