

CSCI 261 Exam 1 Review Topics

This is **not** a complete list, but these are the big, critical ideas you should know.

Simple Applications in C++: Syntax of a basic `int main()`; how to declare variables (`bool`, `char`, `int`, `string`, `double`); how to declare variables with initial values; how and when to use `const` variables; garbage values (what one variable type will **not** have a garbage value?); literal values; what `char` variables hold, and the ASCII table's most important property: digits, uppercase, lowercase are increasing and sequential; how `char` data is treated special by `cout` and `cin`; postfix operators (`x--`); prefix operators (`++y`); writing equations in C++; special assignment operators (`*=`); printing to `cout` with fixed decimals (`.precision()`, `.setf(ios::fixed)`); using math functions (`sqrt()`, `pow()`, ...) from `#include <cmath>`; the perils of integer division & how to cast around them using `double(x)`. Computational underflow (\Re computations) and overflow.

Boolean Expressions: Logical operators `&&`, `||`, and `!`; relational operators (`>=`, ...); short-circuiting of logical combinations; automatic conversion of numbers to Boolean values and vice-versa; why `2 <= x <= 1` is always true.

Selection Statements: The three different forms of a C++ "statement"; `if` statements; `if-else` statements; `if-else-if- ...` nested control structures; terse `if` statements (`if(x>y) x++`); terse `if-else` statements.

Loops: The differences and similarities between all looping structures (`for` and `while`); rewriting `while` ↔ `for`; rewriting `for` ↔ `while`; scope of variables declared within looping structures. The `break` keyword; ∞ -loops.

File and Stream I/O: OOP vocabulary: class, object (or instance), member functions, and the "dot" member operator; what is "returned" from `cout << x`? what is the side effect of this operation? Opening files with `ifstream` and `ofstream` object declarations; understand C++ conversion (f)stream variables to Boolean values; testing for open failure (`!infile`); use of `exit(1)` on failure; how to read in multi-variable data records & check for failure¹; when and how to call `.close()` for opened files; input stream parsing rules for `bool`, `char`, `int`, `double` and `string` variable types; `.precision()`, `.setf(ios::fixed)`, and `.width()`; `::left` and `::right` justified printing in a fixed width field; why the `'\t'` character sucks for table formatting.

1d Arrays: How are arrays declared; explicit versus implicit declaration of array sizes; implicitly declared array values; explicitly declared array values; what is an element's offset? Why are element offsets important in C++? C++ programmers start counting at what number? How are array elements iterated through with a `for` loop; when used with arrays, how are `[]` interpreted differently by the compiler depending on code context.

2d Arrays: How are arrays declared; explicit versus implicit declaration of array sizes; implicitly

¹`while(infile >> x >> y >> z)`

declared array values; explicitly declared array values; how are rows and columns interpreted in 2d arrays.