

Math 173 Course Content and Objectives

Lecture: <i>nmga amc cbg a , M g c cdca am c bc a gn gn</i>	Hours Per Topic	Lecture: <i>a gn c cc nm nm dn c a gn c co gge anel ggc m am c ,'</i>
Introduction to the C++ language: History of C++, portability and standards, mechanics of creating a program. Variables, const, floating-point numbers, C++ arithmetic operators, cout, and cin.	2	Identify basic C++ standards. Create variables and constants of various types. Use cout and cin to query for user input.
Arrays and multidimensional arrays, enumerated types, strings and std::string, and std::vector. Introduction to the debugger.	5	Write a program that computes grade point average.
Expressions, statements, and operators: All major C++ operators, including bitwise operators, bitwise arithmetic, and using bitmasks.	4	Use bitwise operators and bitmasks to encode and decode data. Decide which bitwise operators to perform a given operation.
Controlling program flow using loops: for loop, while loop, do-while loop, nested loops, multidimensional arrays, and advanced debugging features. Branching statements and logical operators: The if statement, logical expressions, the switch statement, the break and continue statements.	5	Write a program or programs that use for loops and while loops. Write a program that uses nested loops to populate a two-dimensional array. At least one of these programs should model physical motion or some other physical phenomenon.
Functions: Arguments, passing by value, passing by reference, relationship with arrays, relationship with two-dimensional arrays, and an introduction to recursion. Advanced function topics: Inline functions, reference variables, default arguments, and function overloading.	5	Apply recursion to compute the Fibonacci numbers.
Pointers, pointer arithmetic, and dynamic memory allocation through new and delete.	4	Use pointers and dynamic memory to write code that adapts to user input of varying size.
Objects and classes: Procedural versus object-oriented programming, abstraction and classes, data encapsulation, constructors and destructors, the this pointer, arrays of objects, class scope, and abstract data types. Differences with structures.	4	Use classes to model a Human object.
Inheritance: public, private, and protected inheritance, inheritance and method overloading,		

terms of purpose and intended use. Creating programs written in Java.		
Final examination.	2	Final examination.

Lab

Lab: M d c cdca am c bc a g gñ nmø am c cb d a ',	Hours Per Topic	Lab: c a gñ c cc nm nt m dn a gñ c co g d e anel gg c m am c ,'
Input, output, and variable types.	2	Write a program that computes geometric properties of circles and spheres based on a user inputted radius.
Using arrays in computations and working with std:string.	3	Write a program that computes the dot product of two vectors. Write a program that uses string arithmetic.
Using operators and bitwise operators.	3	Write a program that uses bitwise operators and bitmasks to encode and decode data.
Conditional statements and loops.	3	Write a program that computes a user-specified number of prime numbers.

Functions and recursion.