

Math 173 Course Content and Objectives

COURSE CONTENT AND SCOPE - Lecture: Outline the topics included in the lecture portion of the course (<i>Outline reflects course description, all topics covered in class</i>).	Hours Per Topic	COURSE OBJECTIVES - Lecture: Upon successful completion of this course, the student will be able to... (<i>Use action verbs - see Bloom's Taxonomy for 'action verbs requiring cognitive outcomes.'</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, the order of construction and destruction. Using namespaces to organize code.	4	Create an Adult and Child class that inherit from a Human class.
Polymorphism: Virtual functions, pure virtual functions, creating an interface through an abstract base class. Preprocessor directives and creating header files.	4	Select for appropriate behavior and functionality in a Shape class by implementing polymorphism.
Operator types and operator overloading. Casting operators.	5	Implement operator overloading in a Fraction class.
File handling including file input and output. Exception handling including try and catch.	5	Use file handling functionality to read and write from a file.
Comparison of Java and C++, including major and minor syntax and language differences. Explanation of the Java language in	5	Take previous code from the course written in C++ and rewrite it using Java.

terms of purpose and intended use. Creating programs written in Java.		
Final examination.	2	Final examination.
Total:	54	
Total Lecture Hours In Section I Class Hours:	54	

Lab

COURSE CONTENT AND SCOPE - Lab: Outline the topics included in the lecture portion of the course (<i>Outline reflects course description, all topics covered in class</i>).	Hours Per Topic	COURSE OBJECTIVES – Lab: Upon successful completion of this course, the student will be able to...(Use action verbs – see Bloom's Taxonomy for 'action verbs requiring cognitive outcomes.')
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.	3	Write a function that models a power function.
More recursion, processing keyboard input, using a graphics API.	3	Write a program that navigates a randomly generated maze. Write a program that moves a graphic object according to user input.
Classes, constructors, destructors and object-oriented design.	3	Write a program that models bouncing balls.
Advanced class topics.	3	Write a program that models stacking blocks.
Inheritance, polymorphism, and abstract base classes.	3	Write a program that models rotating Tetris blocks.
Operator overloading and casting operators.	3	Write a Fraction class that allows for all the usual operations as well as conversion to std::string and double.
Java for C++ programmers.	2	Take one lab from the semester and rewrite it using the Java language.
Final project.	5	Create a program that utilizes all the material from the course, particularly, sound object-oriented design. This program should require significant planning and documentation.
Total:	36	
Total Lab Hours In Section I Class Hours:	36	