

MATH 211 Computational Mathematics I

1. Catalog Description

MATH 211, 212 Computational Mathematics I, II (4) (4)

Fundamentals of procedural programming in C/C++ and selected applications to problems in integral and differential calculus, matrix analysis, geometry, and physics. 4 lectures. **MATH 211** prerequisite: MATH 141 or consent of instructor. **MATH 212** prerequisite: MATH 211.

2. Required Background or Experience

Math 141.

3. Learning Objectives

The student should:

- a. Understand basic concepts of operating systems and programming languages.
- b. Be able to write and debug programs using the procedural part of C++, including input/output, flow control statements and functions.
- c. Learn how to use programs to solve a variety of mathematical problems taken from algebra, elementary calculus and geometry.

4. Text and References

A text containing detailed descriptions of programming projects will be prepared by the Mathematics Department. Students will also use a C++ reference.

5. Minimum Student Materials

Paper, pencils, notebook and access to university computing resources.

6. Minimum University Facilities

Classroom with ample chalkboard space for class use.

7. Content and Method

1. Compiling and running a program	4
2. Basic data types, variables and assignment statements	4
3. Flow control statements, input and output	8
4. Arrays	4
5. Functions and structured programming	6
6. Debugging	4
7. Recursion	<u>4</u>

Total 34

8. Methods of Assessment

The primary method of assessment is assigned programming projects. In addition, written exams on programming and problem-solving will be used. The emphasis should be on the whole problem-solving process, not just on writing correct programs. All instructors will use the same assessment methods, though individual instructors may weigh the assessment methods differently. The learning objectives are listed below. The numbers in parentheses following each objective indicate the appropriate assessment method according to the scheme 1 = programming projects, and 2 = written exams.

The student should:

- a) Understand basic concepts of operating systems and programming languages. (1,2)
- b) Be able to write (1,2) and debug (1) programs using the procedural part of C++, including input/output, flow control statements and functions.
- c) Use programs to solve a variety of mathematical problems taken from algebra, elementary calculus and geometry. (1)