Math 475 Advanced Mathematical Software Programming (4 semester hours)

Course Description:

Students will learn fundamental concepts of computer programming and write software to implement a variety of mathematical algorithms, manipulate large amounts of data, test conjectures, and make abstract mathematical concepts concrete. Programming concepts include input versus output, data structures, local and global variables, switch statements, iteration, recursion, halting conditions, modularity, debugging, and algorithm analysis. Programming projects may vary with instructor, but could include topics from enumerative combinatorics, graph theory, group theory, linear algebra, and number theory.

Prerequisites: MATH 345 and Math 375 with a “C” (2.0) or better, and senior standing or consent

Learning Outcomes:

1. Choose a suitable data structure a given mathematical construction (list, matrix, etc.) and describe the basic operations to be performed.
2. Use logic, iteration, recursion, and other techniques when reading and writing software.
3. Write pseudocode for an algorithm that describes its inputs and outputs, its logic and procedure. Implement the algorithm in software using logic, iteration or recursion as appropriate. Write the software in a clear style, commenting the code to convey its purpose and design.
4. Evaluate the steps of a program step by step, interpret error messages and identify their source, correct errors and revise the program as necessary.
5. Analyze and gather computer-generated data. Make and test conjectures. Solve a problem by brute force that is too complicated to work out by hand. Display results in a clear and organized fashion.
6. Describe both qualitatively and quantitatively the relationship between input size and either runtime or memory usage. Critique an algorithm, suggest improvements, implement the improvements in software.
7. Describe a problem to be solved, its method of solution, and the results obtained. Present projects in both written and oral form.
Content Outline:

Every student will have access to a computer equipped with mathematical software (for example, Mathematica). The instructor will work through a series of increasingly sophisticated programming examples, introducing techniques as the need arises, and supply students with starter code that they can modify as necessary. Students will develop a repertoire of advanced programming skills and apply them to solving mathematical problems and both making and testing conjectures. Graduate students enrolled in the course will be asked to submit an additional project.

Textbook/Related Readings/Materials: