

# Math 4984: Spring 2011

## Programming and Mathematical Problem Solving

### Prerequisite Math 2214 / Corequisite Math 2224

**Instructor: Eric de Sturler**

Office: 544 McBryde Hall

Email: sturler (at) vt (dot) edu

Web page: <http://www.math.vt.edu/people/sturler/index.html>

Phone: (540) 231 5279

**Time: Tuesday/Thursday, 8:00-9:15**

**Place: Torgersen Hall 1020**

**Office Hours: Tuesday 915-1115 am, Thursday 330-530 pm, 544 McBryde**

#### General Description

This course is being designed to serve as an introductory programming course for Mathematics majors. We will study basic programming techniques in the context of problems typically encountered by mathematicians. We start with procedural programming and towards the end of the course move to object-oriented programming techniques. The main programming language for the course is Matlab. Time permitting we may discuss Mathematica and solving problems for Mathematica is particularly effective.

The course may be used to satisfy the computer science requirement on the four Math Major option check sheets. It may not be used to fulfill 4000-level Math requirements.

**Prerequisite:** Math 2214.

**Corequisite:** Math 2224.

#### Text and Software

Insight Through Computing, Charles Van Loan and K.-Y. Daisy Fan, SIAM 2010

Additional material is available on the scholar site (under Resources).

Students are required to have access to Matlab (and possibly later Mathematica) in order to perform computations, edit, and run programs. Matlab is available for \$39 and Mathematica is available free to all Virginia Tech students through the Virginia Tech Computing Center. You may download them at <http://www2.ita.vt.edu/software/student/>. For those who do not wish to install these programs on their computers, they are available on all computers in the Math Emporium where they can be used without charge.

#### Syllabus

- Basic procedural programming, including simple programming examples, numerical and non-numerical but focusing on mathematical problems, to demonstrate concepts,
- Brief overview of computer architecture focused on what is needed for effective programming and debugging, how computers store programs and data, how do computers execute programs
- Types, declarations, expressions and operators. Control flow,
- Functions (subroutines) and program structure,
- Data structures,
- Important theoretical programming concepts (structured programming, top-down design, step-wise refinement).
- Graphics and visualization
- Write programs for a number of numerical and non-numerical problems using the procedural approach, from simple programs to programs requiring multiple subroutines and complex data structures, use graphics/visualization.
- Object-oriented programming
  - Why object-oriented programming?
  - Main concepts, abstraction, encapsulation (information hiding), modularity, hierarchy, classes (types),
  - Classes, objects, and functions – implementation in Matlab,
- Write programs for a number of numerical and non-numerical problems using the object-oriented approach. Adapt earlier programs to object-oriented programming style.

#### Time permitting:

- Solving mathematical problems using Mathematica
- Description of Mathematica,
- Demonstrate where the programming model has advantages
- Mathematica programming/functional programming

#### Grade

The grade will be determined by homework and projects. Assignments during the semester will make up 70% of the grade. The final project will take the place of the final exam and is worth 30% of the grade.

Homework will be assigned weekly or biweekly. The maximum number of points awarded for the assignment will determine the “weight” of each assignment. That is, a 200-point assignment is worth twice as much as a 100-point assignment, etc. All grades will be posted on the course Scholar site.

An average grade of 90% or better will guarantee an A, 80% at least a B, 70% at least a C, and 60% at least a D.

#### Assignments

All assignments will be posted on the course Scholar page. The time and date that the assignment is due will be posted with the assignment. Late assignments are not accepted unless extenuating circumstances apply (typically a doctor’s note or note from the dean of student’s office is required). Again for special circumstances you can apply for some extra time, **in advance**, with the instructor.

#### Electronic Submission of Programs

Most of your assignments will require the electronic submission of programs, possibly with output and additional explanation through the class Scholar page. You must follow the following conventions:

1. Program names and output files must contain (1) the assignment number and (2) your name (abbreviated if necessary) separated by an underscore, that is, P317\_desturler.m and (if output is generated that should be provided) P317\_desturler.out.
2. In the body of the program you must include a comment header that includes (1) your full name, (2) the assignment number, and (3) the year and semester.
3. Programs and functions must be written in such a way that, when downloaded all into a single directory, the program runs as intended **without any additional effort. If you are required to test the program for a number of inputs and/or parameters you must provide a script that carries out these tests and submit the required output with the homework.** If it is not obvious how to run a program or function, you must provide an additional file called ‘Readme’ that explains how to run the program or function, and you must provide a script (and input files if necessary) that runs the program or function as intended (providing example input, parameter values, etc.).

In general, your programs may not be accepted unless you have conformed to these rules and conventions, and you will lose points if any or some rules are not satisfied.

#### Documentation of Programs

Especially in academic science, documentation of programs is crucial to transparency and reproducibility of results. In a course of programming, good documentation will help you understand what you are doing (both when you are first doing an assignment and when you go back to use it later). All programs must be clearly documented. At least 10% of the grade on all m-files and notebooks will be based on documentation.

#### Special Needs

Any student with special needs or disabilities should schedule a time to see me as soon as possible.