Math 131 - Mathematics Through Advanced Software - Fall 2010
Course Syllabus

SUMMARY INFORMATION
Instructor: Jeff Hamrick, Ph.D., CFA, FRM
Office: Ohlendorf 318
Office Hours: 3:00-3:50 p.m. M, 10:00-10:50 a.m. W, and 3:30-4:50 p.m. Th
Office Phone: 901/843-3253
Email Address: `hamrickj@rhodes.edu`

Class Location: Barret 035
Class Time: 6:00-8:00 p.m. on Wednesdays

ON COMPUTER ALGEBRA SYSTEMS. A **computer algebra system** is a software program that facilitates symbolic mathematics. Though the main purpose of a computer algebra system is manipulation of mathematical expressions in symbolic form, over the past few decades computer algebra systems have evolved to include formidable numerical and data analysis capabilities.

By "symbolic manipulation," I mean simplification of certain expressions to some standard form (with user-designated constraints often being possible). Computer algebra systems permit the user to substitute symbols or numerical values in place of certain expressions, as well as to make changes in the form of those expressions (partial and full factorization, representation as partial fractions, rewriting trigonometric functions in terms of exponential functions, etc.). Many standard tasks of calculus are well-handled by computer algebra systems: the symbolic computation of limits, derivatives, integrals, and infinite sums; symbolic constrained and unconstrained local and global optimization; and the calculation of Taylor expansions of analytic functions. Increasingly, computer algebra systems are being used to power automated theorem-proving and theorem verification, two tasks that are central components of the growing field of **experimental mathematics**.

Mathematica is just one example of a computer algebra system, but there are many others, including Maple, Derive, Reduce, MuPAD, Magma, Axiom, and Maxima. Though proprietary (and, arguably, expensive), *Mathematica* is the computer algebra system most frequently used at Rhodes College (and, in fact, at many colleges, universities, corporations, and research laboratories). It was originally conceived by Stephen Wolfram, a MacArthur fellow, physicist, and author of *A New Kind of Science*, as a computer program called SMP (symbolic manipulation program). *Mathematica* 1.0, the successor to SMP, was released in 1988 and has improved incrementally with new major releases in 1991, 1996, 1999, and 2003. With the release of *Mathematica* 6.0 in 2007, *Mathematica* experienced enormous improvements in system stability, software documentation, dynamic interactivity, and data computability. In 2009, Rhodes College upgraded its ten-year-old *Mathematica* 4.0 license to a new *Mathematica* 7.0 license. Today at Rhodes College, *Mathematica* is used for undergraduate and faculty research, projects in upper-level courses in mathematics and the natural sciences, and instruction in Math 115 (applied calculus).

ON COURSE GOALS. Any student who successfully completes this course should:

- Understand the difference between the front end and kernel in *Mathematica*;
- Be able to determine the heads of expressions;
- Control the precision and accuracy of numbers;

- Understand the difference between functional and procedural programming;
- Use prefix and postfix notation to make *Mathematica* code more readable;
- Understand how to use *Mathematica* to repeatedly apply a function;
- Manipulate (and apply functions to) matrices and sparse arrays;
- Collect, exclude, and count items that do and do not match particular patterns;
- Replace parts of expressions, perhaps repeatedly;
- Compare, manipulate, remove, and replace strings;
- Match, search, and tokenize text;
- Plot functions in Cartesian and polar coordinates;
- Create contour and density plots;
- Import, export, process, compress, and manipulate both images and audio files;
- Create and transform algebraic expressions like polynomials; and
- Numerically and symbolically compute limits, derivatives, finite and infinite series, integrals, and difference equations.

In general, this course will prepare successful students to use a computer algebra system (and, in particular, *Mathematica*) to support intellectual investigations in mathematics, physics, economics, finance, physics, chemistry, biology, and related fields.

ABOUT ME. My name is Jeff Hamrick. I'm an assistant professor in the Department of Mathematics and Computer Science at Rhodes College. Please call me Jeff. My office is located in room 318 of Ohlendorf Hall. I will hold office hours from 3:00 p.m. - 3:50 p.m. noon on Mondays, 10:00 a.m. - 10:50 a.m. on Wednesdays, and 3:30 p.m. - 4:50 p.m. on Thursdays during the spring semester. My office number is 901/843-3253 and my e.mail address is `hamrickj@rhodes.edu`. Please stop by my office anytime. If you're unable to make my office hours, let me know and we may be able to schedule an appointment at an alternate time.

ABOUT YOU. You should be hard-working and enthusiastic about learning! This course features a basic introduction to *Mathematica*, a particular computer algebra system, and applications of *Mathematica* to topics or issues in mathematics and the natural sciences. You should have a strong working knowledge of high school algebra to succeed in this course. Ideally, you will also have had at least one semester of calculus, though the first calculus course is not necessary if you are willing to learn some ideas from linear algebra and the second course in calculus as we explore *Mathematica*.

ABOUT US. We will meet to talk about *Mathematica* and using it to study mathematics on Wednesdays from 6:00 p.m. - 8:00 p.m. during the fall semester. We will meet in Barret 035. I suggest (but do not require) that you use *The Mathematica Cookbook*, by S. Mangano (ISBN: 978-0-596-52099-1). We will more or less cover the first through eleventh chapters in this textbook, though we will not necessarily cover every part of every chapter (in particular, our treatment of audio files and images will be somewhat light). There are several copies of this book available for purchase at the college bookstore. Additionally, you may want to consider going to my public folder and swiping a copy of *Mathematica Programming: An Advanced Introduction* by L. Shifrin. This document is freely available on the Internet.

ON ATTENDANCE. Attendance is expected in this course but is neither required nor rewarded. If you miss a class, you will find some *Mathematica* scratch work in the public folder for the course. However, you should still try to go over the lecture material you missed with a colleague in the class.

ON HOMEWORK. I will assign a set of *Mathematica*-based exercises during every class period. Each set of exercises will always be due, in its entirety, by 3:00 p.m. on Wednesday of the **following** week. **You must work on the starred exercises by yourself—you may use the Help browser and *Mathematica* books, but you may not consult with colleagues and/or faculty members.** To clarify: you must turn in all of the problems, but you must work on the starred problems by yourself.

I expect you to work on the assigned exercises very frequently but for short periods of time. (Starting any software- or programming-based assignment the night before is **never** a good idea.) To turn in a *Mathematica* assignment, you will need to drop a *Mathematica* notebook into a specially-designated folder inside of my public folder. **I will not accept late homework assignments under any circumstances.**

ON THE FINAL COURSE PROJECT. You must choose any one of the following three final projects for the course.

- **Take and pass the *Mathematica* student certification test at the Foundations Level or higher.** For more information about this free program supported by Wolfram Research, go to

  `www.wolfram.com/services/certification`.

- **Author, submit, and successfully publish a demonstration at the Wolfram Demonstrations Project.** Your demonstration must be reasonably original (i.e., it should not already be published at the Wolfram Demonstrations Project web site). It need not be related to mathematics, though it might be. Submissions to the Wolfram Demonstrations Project are peer-reviewed both for the correctness and compactness of the submitted *Mathematica* code and correctness and value of the content being submitted. For more information about creating and publishing a demonstration, go to

  `demonstrations.wolfram.com`.

- **Use *Mathematica* to significantly augment a class project or research project for some Rhodes-based activity (group research in the natural sciences, a capstone project, etc.) or class.** Your proposed augmentation must be for a course in which you are currently enrolled or for a project with which you a currently involved. You may propose, with the consent of both me and the faculty member teaching the other course, a new project (i.e., you do not have to augment an assignment for the other course—your project could be related to the other course but could be of your own design). If you are augmenting a project or assignment from another course, the augmentation must be significant and must be beyond the scope of what the other course's instructor was expecting you to do. If you choose to exercise this particular option, please consult with me early and often.

The final project (proof that you have passed the certification exam, the web address of your published demonstration, or the augmentation of some project for another course or activity) is due by 3:00 p.m. on the last day of the final examination period (December 15).

ON EXAMINATIONS. There will be no examinations in this course. I will determine if you are independently developing your *Mathematica* abilities by your performance on the starred homework problems, on which you are not permitted to collaborate with colleagues, faculty members, or other third parties.

ON GRADING. I've noticed that students are too focused on grades, to the great detriment of their own learning. If students put as much effort into actually learning material as they did worrying about their grades, their performance would be much better. Nevertheless, part of my job is to assign grades fairly and in a manner that reflects the high academic standards at Rhodes College. In this class, we will use the standard ten-point scale. "Plus" or "minus" grades will be assigned to students with grades close to the extremes of each ten-point bracket (plus or minus three points from the boundary of each bracket). In other words, a final grade of 93-100% is an A, 90-93% is an A-, 87-90% is a B+, 83-87% is a B, etc.

Your grade in this course will be computed according to the following weights:

Component	**Weight**
Mathematica Assignments & Exercises	80%
Final Course Project	20%

ON CHEATING. In this class, we will adhere to the provisions of the Rhodes College Honor Code. In general, I encourage you to work on *Mathematica* assignments (**except for the starred problems**) with colleagues **but you may never copy work from colleagues verbatim or copy their work and merely paraphrase it**. In general, if the Rhodes College Honor Council concludes that a student in this course has violated the honor code, I will adhere to the recommendations of the honor council. However, I reserve the right to lower a student's grade in this course if I sincerely believe that an infraction has occurred.

COURSE OUTLINE. A *tentative* outline of the topics that we will cover (and the particular days on which we will cover these topics) is as follows:

Date	**Topic(s)**
August 25	*Mathematica* basics, functions, numbers, precision, accuracy, and intervals
September 1	more on functions: mapping functions, holding arguments, threading, indexing
September 8	still more on functions: pre- and post-fixing, iteration, composition, inverses
September 15	lists, sorting, matrices, sparse arrays
September 22	deeply nested lists, graphs, graph algorithms, random graphs
September 29	pattern matching, collecting items, counting items, substitution rules
October 6	manipulating patterns with patterns, semantic/unification pattern matching
October 13	comparing strings, replacing characters, extracting substrings, searching text
October 20	plots in Cartesian/polar coordinates, parametric plots, mixing plots, legends
October 27	text annotation, three-dimensional plots, contour plots, polyhedra
November 3	extracting information from images, enhancing images, edge detection
November 10	creating musical notes, scales, melodies, chords; import and export
November 17	algebraic equations; polynomials: factoring, expanding, division
December 1	limits, piecewise functions, power series, differentiation, integration
December 8	differential equations, optimization, vector calculus, infinite series