Analytical Programming – MRM 8000
Summer/Fall 2015
(CRN 87543/88754)

Instructor Information

- **Instructor:** Dr. Daniel Bauer
- **Office:** Room 1126, Robinson College of Business (35 Broad Street)
- **Office Hours:** By appointment (just send me an email)
- **Email:** dbauer@gsu.edu. **Skype:** danielbauer1979. **Phone:** 404-413-7490

Class Information

- **Location:** GSU Buckhead Center, Room 404
- **Time:** August 3-14, Monday-Friday, 10:00 am - 1:00 pm (Class starts on 08/03/2015)
- **Prerequisite:** Calculus and some basic statistics. Robinson College of Business Computing Skills Prerequisites (CSPs) 1-4, 6, 7 (see http://robinson.gsu.edu/rcbonline/csps.htm)
- **Website:** D2L Brightspace – https://gsu.view.usg.edu/

Catalog Description:

This course is an introduction to analytical programming and numerical methods. The objective is to learn how to develop algorithms and computer code for scientific computing with a view towards the practical application of mathematical models in risk management, insurance, economics, finance, and related fields. The course emphasizes the principles and numerical techniques used to turn algorithms into reliable and efficient computer programs.

Course Objectives

This graduate-level course is an introduction to programming and numerical methods. The main objective is to learn how to write and compile computer programs in a variety of programming languages, to pick up programming basics but also some advanced concepts such as object-oriented programming, and to implement numerical algorithms with a view towards applications in risk management and analytics. By completing this course, successful students will be able to:

- Prepare, compile, and run basic computer programs in a variety of programming languages including C/C++, Java, Python, R, Stata, and VBA.
- Understand the principles of object-oriented programming (OOP), and apply basic OOP principles.
- Implement advanced algorithms for solving mathematical or statistical problems in a structured fashion in R.
- Broadly understand and be able to apply methods from numerical mathematics, including quadrature (numerical integration), numerical solution of non-linear equations, optimization, and random-number generation.
High-level languages such as Matlab, R, or Stata already contain a variety of implemented statistical and mathematical procedures. In this course, rather than relying and discussing these advanced toolboxes, we focus on the problem of turning algorithms into code. So the intended audience is those who want to make tools, not just use them.

**Methods of Instruction**

**Classroom Procedures**

The course material is presented in lecture form and we will go over a variety of examples. Students should bring their laptops to class to take notes but also to be able to run some example programs.

**Attendance Policy**

Attendance is not formally taken. However, it is strongly suggested that students do not miss class, and I reserve the right to adjust grades based on attendance and/or class participation.

For students that are not able to attend class, we will provide an online option. In case you need to take advantage of this option, please email me. We will provide information on logistics etc. shortly before the beginning of class. Note that this option is only for special circumstances in case someone has a very good reason not to attend class in person (e.g., because you are not arriving in Atlanta until after the class begins – in contrast, reasons such as having to walk your parents’ dog every day are not acceptable...) I will ask for proof such as your airline ticket. If you just email me “to check” if it is possible even though you do not have a very good reason, the answer will be a grouchy and resentful NO!!! Furthermore, note that even in case you cannot make it to class, you are responsible and required to submit your assignments and project work by the given deadlines – which always refer to Atlanta time (even if that happens to be the middle of the night in your time zone!) In particular, you will need Internet access. I will drop one of your homework scores, so even in case you will have to miss one homework assignment due to travel etc., that does not present a reason for special arrangements.

**Class Material**

We will rely on a variety of textbooks and online resources (see below). Since there are several options of programming languages you can use for certain assignments (although there are problem-specific restrictions), there is not one particular textbook you need to buy. However, I can give recommendations what makes sense in case you want to buy a book. Lecture slides and example code will be posted online.

**Homework and Projects**

There will be a homework assignment every day due thirty minutes before the next class. These assignments will typically be fairly straightforward, and their intention is to make sure you understood and digested the material presented in the foregoing class.

There will be one larger mid-class project. I will give out the assignment on the last day of the first week (Friday), and the assignment will be due thirty minutes before the first class of the second week (Monday).

There will also be a final project, which I will give out on the last day of class, and it is due in the beginning of the subsequent week.

All work has to be done on an individual basis. You can discuss the assignments with your classmates, but in the end everyone has to complete the assignment by herself/himself. In particular, if you use code written by another student (or an online source without disclosing it properly), I will impose severe penalties. Cheating will not be tolerated!

**Grading Criteria**

The mid-class and final project each will count for one third of your final grade, and the homework will count for the remaining $33^{1/3}\%$. Homework will be graded in a cursory manner, and I will waive one your scores (so that not submitting one of your assignments will not affect your grade).

Grades will be awarded on a +/- basis, and the following guaranteed scale applies. Grades may be moved upward based on difficulty, but not downward:
Remarks

- Students exhibiting disruptive behavior, including talking, sleeping, talking on cell phones or disturbing other students will be asked to leave.

- Please advise the instructor if you have a documented disability that needs to be accommodated.

- As members of the academic community, students are expected to recognize and uphold standards of intellectual and academic integrity. See the University’s policy on Academic Honesty (Section 409, http://www2.gsu.edu/~wwwfhb/sec409.html) for details.

Outline and Schedule:

The class will be separated into two parts.
In the first week, we will discuss programming basics in the context of a “lower-level” general-purpose programming language. In particular, we will become familiar with C/C++ and Python as two of the most popular languages (together with Java more than 60% of developed code in 2015, http://blog.codeeval.com/codeevalblog/2015; C++ has high relevance in the financial services industry, and Python is on the rise in analytics). We will focus on programming basics such as variable declarations, conditional statements, loops, functions, arrays, etc. but we will also touch on more advanced concepts such as object-oriented programming (OOP).

In the second week, we will discuss the implementation of numerical algorithms with a focus on risk management and analytics applications. While we will take advantage of the concepts acquired in the first part of the class, we will rely on one of the “higher-level” languages that were created for these purposes. Due to its relevance in statistics, analytics, and quantitative finance, we focus on R. Here we will touch on some concepts from numerical mathematics, particularly linear algebra routines, quadrature (numerical integration), numerical solution of non-linear equation systems, numerical optimization, and random-number generation. However, the focus is on implementing algorithms rather than the underlying mathematics.

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday, 08/03/2015</td>
<td>“Pick your weapon I:” Introduction to computing and to programming languages. Introduction of C/C++, Java, Python, and VBA.</td>
</tr>
<tr>
<td>Tuesday, 08/04/2015</td>
<td>“The Basics I:” Variables, declarations, operators, statements, arrays, strings, IO.</td>
</tr>
<tr>
<td>Wednesday, 08/05/2015</td>
<td>“The Basics II:” Control flow, if-else statements, loops, recursion.</td>
</tr>
<tr>
<td>Thursday, 08/06/2015</td>
<td>“Modularization and Abstraction the old way:” Functions/modules, pointers.</td>
</tr>
<tr>
<td>Friday, 08/07/2015</td>
<td>“Abstraction the new way:” Introduction to object-oriented programming (OOP), classes, overloading. Mid-class project assigned.</td>
</tr>
<tr>
<td>Tuesday, 08/11/2015</td>
<td>“Programming in R:” Using R as a programming environment.</td>
</tr>
<tr>
<td>Wednesday, 08/12/2015</td>
<td>“Basic numerical analysis:” Getting comfortable, vectors and matrices (some linear algebra) and statistics.</td>
</tr>
<tr>
<td>Thursday, 08/13/2015</td>
<td>“Some numerical recipes:” Solving non-linear equations, numerical integration, optimization, random number generation and Monte Carlo, finite differences and differential equations.</td>
</tr>
<tr>
<td>Friday, 08/14/2015</td>
<td>“Wrap-Up:” More numerical recipes, some problems and pitfalls. Final project assigned.</td>
</tr>
<tr>
<td>Monday, 08/17/2015</td>
<td>8:30am-11am: Available for questions.</td>
</tr>
<tr>
<td>Tuesday, 08/18/2015</td>
<td>6pm: Final project due.</td>
</tr>
</tbody>
</table>

Note that this course syllabus provides a general plan for the course; deviations may be necessary.
Resources: \(^1\)

“Classical” programming text- and reference-books


Textbooks for numerical methods


Helpful online resources on programming languages (mostly free)

- Python: [https://www.python.org](https://www.python.org).
- The R Project: [http://www.r-project.org](http://www.r-project.org).
- Matlab: [http://www.mathworks.com/products/matlab/](http://www.mathworks.com/products/matlab/) (Matlab is not free. There are trial versions, though, and Matlab is available on GSU machines.) Octave (a free version of Matlab, does not have the same libraries etc., but should execute basic Matlab code): [http://www.gnu.org/software/octave/](http://www.gnu.org/software/octave/).
- Stata: [http://www.stata.com/order/new/edu/gradplans/student-pricing/](http://www.stata.com/order/new/edu/gradplans/student-pricing/) (also not free, but reduced pricing for students). SAS is available for free to GSU students: [http://technology.gsu.edu/technology-services/it-services/](http://technology.gsu.edu/technology-services/it-services/).

Other online resources


---

\(^1\)No need to buy anything!