Syllabus

Course Information

Name:	EMSE 6574: Programming for Analytics, Section 10
Semester:	Fall 2019
Meeting Time:	6:10pm - 8:40pm
Location:	Phillips Hall 108

Instructor Information

Name:	John Paul Helveston
Campus Address:	Science & Engineering Hall, Office 2830
Campus Phone:	+1 (202) 994-7173
Email:	jph@gwu.edu
Office Hours:	TBD

Course Description

Official GW Bulletin Description

Introduction to programming using the R programming language; topics include functions, conditionals, loops, strings, file input/output, plotting, coding style, efficiency, Monte Carlo methods, packages, and shiny applications.

Unofficial Description

This course provides a foundation in programming using the R programming language. Emphasis will be on producing clear, robust, and reasonably efficient code using top-down design, informal analysis, and effective testing and debugging. Throughout the course, students will primarily work on individual programming assignments to help practice coding skills. Students will be assessed through quizzes, exams, and a final project. Teaching will involve interactive lectures and recitations with a lot time spent live coding. At the end of the semester, students will demonstrate mastery of the course's topics by working on a 3-week team project. This course assumes no prior programming experience and will prepare students for higher level courses in data analytics.

Prerequisites

There are no prerequisites, and no prior programming experience is necessary to succeed in this class.

Learning Objectives

Having successfully completed this course, students will be able to:

- Read, write, test, and debug code using top-down design.
- Use computational thinking to solve problems.
- Explain the efficiency of algorithms by predicting the Big-O run time of code.
- Design and write an `R` package and deploy it on Github.
- Design and write an `R` shiny application and deploy it on Github.

Required Texts & Materials

This course does not require any textbooks. All learning materials and resources will be made freely available on the course resources webpage.

Software

All required software for this course is freely available on the web. This includes:

- R (version 3.6.0 or later), which can be downloaded from The Comprehensive R Archive Network (CRAN)
- RStudio (Desktop Version), a free IDE for R, which can be downloaded from RStudio

Go to the course "Getting Started" page for installation instructions.

Other Useful Resources

You will inevitably run into problems while learning R. The best starting point for troubleshooting is Google. In addition, these resources can also be helpful:

- Data Wrangling in R:
  - Wickham, Hadley. "R for Data Science" [free online], [buy on amazon]
- Data Visualization:
  - Healy, Kieran. "Data Visualization: A practical introduction" [free online], [buy on amazon]
  - Wilke, Claus O. "Fundamentals of Data Visualization" [free online], [buy on amazon]
- Handy R "cheatsheets"

Assignments and Grades

Homework Assignments

Homework assignments contain a mix of coding exercises (reading, debugging, and writing code) and written exercises. They generally assess the material taught the week they are assigned, and should take several hours to complete. You are **strongly** encouraged to start the homeworks early (Tuesday is best, and Thursday at the latest!) and come to office hours for help when needed.

Read the Collaboration Policy about collaborating with fellow students on homeworks. While most problems must be worked on individually, some will be marked as collaborative; on these problems (and *only* these problems), you may work on code with other students.

Assignments will be graded based on style (modularity, readability, commenting, etc.) and functionality (correctness on all possible test inputs). Your code should be properly annotated with comments that are well-placed, concise, and informative. Your assignments will be graded by your TA, by automated graders, and at times by your instructor.

Assessments

Quizzes:

Quizzes will be given about once every two weeks immediately at the beginning of class. Quizzes cover material from the previous week and the previous homework assignment. Quizzes are designed to be time-intensive, to test for fluency, and to demonstrate where additional study is needed.

Midterm Exams:

There will be two midterm exams given in class, as noted in the course schedule. Each exam will cover only material presented in the preceding weeks.

Final Exam:

There will be a standard final exam during the final exam period at the end of the semester. This exam will cover material from the entire semester. The final exam is built to allow enough time to attempt all problems.

Class Participation

Regular class attendance is essential. Much of the class time will be spent doing exercises and coding. Multiple absences, inappropriate or unprofessional behavior during class (such as monopolizing discussions or being rude or disruptive), not participating in classroom exercises, and not being prepared for class will result in a lower grade for the class participation component. As a rule of thumb, the participation grade will be assigned according to the following rubric:

Score	Attendance	Classroom
Low	Frequently absent	Rude; disruptive; distracting; monopolizes discussions

Score	Attendance	Classroom
Moderate	Attended most classes, but often arrived late or left early	Takes notes; attentive; occasionally contributes in class discussion / exercises
High	Attends on time and prepared	Takes notes; attentive; regularly contributes in class discussion / exercises; does not dominate conversation; listens and responds thoughtfully to comments made by others

Grading Policy

Standard Grading

Final grades will be calculated as follows:

Course Component	Weight	Notes
Homeworks	30%	Lowest homework grade is dropped
Quizzes	15%	Lowest quiz grade is dropped
Midterm Exams	30%	Lowest midterm grade is half-weighted
Final Exam	20%	Notes
Attendance & Participation	5%	Notes

AMD Grading

This Alternative Minimum Grading (AMG) policy is available to everybody, but is designed specifically for those students who struggle in the first part of the course, and then through sustained hard work and dedication manage to elevate their performance in the latter part of the course to a level that merits passing with a C, even if their Standard Grade might be lower than that.

In addition to the normal grading scale, we will separately compute your grade using the following scale. Students do not sign up for AMG grading. Every student will be considered both for Standard Grading and AMG, and their semester grade will be the higher of the two (where the highest grade via AMG is a C).

To compute your AMG score, first use the following to compute your raw score. If the resulting score is higher than a C, set it back to a C.

Course Component	Weight
Best 5 Homeworks	20%
Best 5 Quizzes	15%
Best Midterm Exam	20%
Final Exam	40%
Attendance & Participation	5%

Unlike the Standard Grade, effort is heavily factored into your AMG score. To qualify for AMG you must put forth sustained effort, which means meeting the following requirements:

- You cannot miss multiple class periods
- You cannot miss multiple assignments or quizzes
- You cannot violate the Collaboration Policy

Grading Scale

Mid-semester and Final grades will be assigned using the following standard scale:

Letter Grade	From	To
A+	97	100
A	93	96.99
A-	90	92.99
B+	87	89.99
B	83	86.99
B-	80	82.99
C+	77	79.99
C	73	76.99
C-	70	72.99
D+	67	69.99
D	63	66.99
D-	60	62.99
F	0	59.99

The course instructors may choose to change the scales at their discretion. You are guaranteed that your letter grade will never become worse as a result of changing scales.

Collaboration Policy

Learning how to program is like learning how to ride a bicycle - to get better, you must practice writing code yourself. Therefore, we have a set of strict rules regarding what kind of collaboration is allowed, what counts as over-collaboration, and what counts as cheating.

This section is long, but you should read all of it thoroughly, as we take collaboration and cheating very seriously in this course. To begin, here's a summary:

- Write your own code.
- Don't look at other people's code.
- Don't let other students look at your code.

Good Collaboration

For all homework assignments, students are encouraged to talk to each other about high-level concepts regarding the problems. This includes:

- Going through existing test cases and discussing why an input results in a certain output.
- Figuring out new input/output pairs for the problems.
- Discussing which general concepts might be useful in solving a problem (conditionals, loops, etc.).

For homework assignments, students may also receive lower-level help from the instructors. This includes everything mentioned above, and also:

- Asking questions about how to build an algorithm for the problem.
- Asking for debugging help with code.

Certain parts of homework assignments may be labeled **Collaborative**. For these parts (and *only* these parts), students may also:

- Sketch out algorithms on a whiteboard together.
- To avoid copying the code, you should write up the solution together, discuss it, then erase the solution, wait a few minutes, and write up solutions individually.
- Help each other debug specific parts of assignment code.
- NOTE: do not 'debug' by telling a friend to try your approach instead! Help them figure out what is actually going wrong.)

Outside of the homework assignments and course assessments, students may also help each other learn to a greater degree. This can include:

- General discussion of course concepts.
- Detailed explanations of example code on the course website.
- Collaboratively solving a practice problem, with any level of co-writing code and co-debugging.

Over-collaboration

Certain actions on homework assignments are considered over-collaboration; this means that a student might have attempted to collaborate with good intentions, but ended up collaborating at a level that became unhelpful for learning. Over-collaboration on homework assignments can include:

- Building 'psuedocode' algorithms together on a whiteboard that are essentially full programs.
- Explaining to a friend how to solve a problem in high-level terms by going through your own program line-by-line.
- Helping a friend debug on a non-collaborative problem.
- NOTE: It is fine to give a friend suggestions if they describe what error is being thrown or which input/output pair is failing, but you may not look at their code/have them read their code out loud.
- Helping a friend debug on a collaborative problem by suggesting they use your own approach to the problem.
- Collaborating with a student on a collaborative problem and then not including their name as a collaborator in the assignment writeup.
- In the case that you have taken this course before, copying your own code from the previous time you took the course.

Over-collaboration results in a warning on the first offense, and a penalty on later offenses.

Cheating

Certain actions within the course are considered cheating - that is, presenting work not completed by yourself as if you wrote it, providing work to others, or receiving unauthorized help.

Cheating on homework assignments can include:

- Copying or stealing any amount of code from someone currently in the class or someone who has taken the class before.
- NOTE: Copying is never okay, whether the code is provided electronically, visually, audibly, or on paper.
- Providing code you have written for an assignment to anyone else in the class.
- Again: never share your code with others in the class, including electronic sharing, showing someone code on your computer, verbally speaking the code, or writing down the code on paper.
- Finding code online and using it in the assignment. One exception: you may use code from the course website.
- Putting code solutions from the course assignments online.
- Receiving code-level assistance from any person not associated with the course.
- Getting someone else to write the assignment code for you.
- Asking questions about the assignments on any online services outside of the course office hours and course Piazza.

Cheating on quizzes, midterms, or the final can include:

- Referring to any external resources while completing the assessment (phones, notes, etc.).
- Copying part of an answer off of another student's paper, even if it is very small.
- Using solutions provided by students who previously took the course.

Cheating results in a penalty on the first offense, and failing the course on the second offense.

Penalties

Penalties are decided by the course instructors, and can vary based on the severity of the offense. Possible penalties include:

- Receiving a 0 on the assignment/quiz in question.
- Receiving a -100 on the assignment/quiz in question.
- Receiving a full letter grade deduction in the course.
- Automatically failing the course.

Penalties may also be accompanied by a letter to the Dean of Student Affairs, again at the instructors' discretion. This can lead to university-level penalties, such as being suspended or expelled.

Plagiarism Detection

Programs are naturally structured, which makes them much easier to compare than hand-written work, and easier to compare than typed essays. We run an automated plagiarism detection system

on all assignments to detect copied code. Even if you make many changes to a copied code, it is *excessively* easy to detect copied code.

In short, if you copy code, we will be able to tell - don't copy code!

Grace Period

Your first year of college is a time when you do a lot of learning. Sometimes, you might make bad decisions or mistakes. The most important thing for you to do is to learn from your mistakes, to constantly grow, and become a better person.

Sometimes, students panic and copy code right before the deadline, then regret what they did afterwards. Therefore, you may rescind any homework submission for up to 24 hours after the deadline with no questions asked. Simply email the course instructors asking to delete the submission in question, and we will do so. Deleted submissions will not be considered during plagiarism detection, though of course they will also not be graded. However, it will always be better to get a 0 (or partial credit) on an assignment than to get a cheating violation!

Course Policies

Late Policy

Each students is allowed **three** late homework submission days throughout the course. You may use them however you wish. No more than two days can be applied toward a single assignment, as homework solutions will be posted two days after the submission deadline. Late days are meant to cover illness, family emergencies, and religious holidays.

Policy on Recording

No student is permitted to record or tape any classroom activity without the consent of the instructor. If a student is disabled and needs to record or tape classroom activities, he/she should contact the Disability Support Services (DSS) to request an appropriate accommodation.

Inappropriate Use of Course Materials

All solutions to homeworks, quizzes, and exams are proprietary. Students are prohibited from posting or selling any such course materials without the express written permission of the professor teaching this course. All other course materials available on the course website are developed open source. You are welcome to post and share them following the licensing guidelines listed in the license page.

Getting Help

Sign up for Piazza HERE

- **Piazza**: Piazza can be used to ask quick questions and receive quick responses without attending office hours in person. This also enables other students to see answers to common questions. Questions on Piazza should be specific and include all needed information. For example, if your code has an error you don't understand, **include the code and the error message** in your question.
- **Office Hours**: Office hours let you ask questions to TAs and professors directly, and can help you understand concepts and debug programs that you're struggling with alone.
- While the University Library is not a stand in for TAs, you can schedule a consultation for general help with Coding, Programming, Data, Statistical, and GIS. See more at https://academiccommons.gwu.edu/writing-research-help

What To Do if the Instructor Does Not Arrive

If the Instructor does not arrive for the class at the designated starting time and has not notified the class of a late starting time or the cancellation of the class, students should wait in the classroom for at least **20 minutes** before departing. One member of the class should be selected to notify the EMSE Department of the Instructor's absence by calling the EMSE Department 202-994-4892 on next business day.

University Policies

University Policy on Religious Holidays

In accordance with University Policy, students should notify faculty during the first week of the semester of their intention to be absent from class on their day(s) of religious observance. Official university policy here: https://students.gwu.edu/accommodations-religious-holidays

- Students should notify faculty during the first week of the semester of their intention to be absent from class on their day(s) of religious observance.
- Faculty should extend to these students the courtesy of absence without penalty on such occasions, including permission to make up examinations.
- Faculty who intend to observe a religious holiday should arrange at the beginning of the semester to reschedule missed classes or to make other provisions for their course-related activities.

Support for Students Outside the Classroom

Disability Support Services (DSS): Any student who may need an accommodation based on the potential impact of a disability should contact the Disability Support Services office at 202-994-8250

in the Rome Hall, Suite 102, to establish eligibility and to coordinate reasonable accommodations. For additional information please refer to: https://disabilitysupport.gwu.edu/

Mental Health Services (202-994-5300): The University's Mental Health Services offers 24/7 assistance and referral to address students' personal, social, career, and study skills problems. Services for students include: crisis and emergency mental health consultations confidential assessment, counseling services (individual and small group), and referrals. https://healthcenter.gwu.edu/counseling-and-psychological-services

Academic Integrity Code

Academic dishonesty is defined as cheating of any kind, including misrepresenting one's own work, taking credit for the work of others without crediting them and without appropriate authorization, and the fabrication of information. For the remainder of the code, see: https://studentconduct.gwu.edu/code-academic-integrity

In addition to the formal code of academic integrity, the instructor expects that students will treat this course with the level of professionalism required in the workplace. Remember that real firms are sponsoring student projects throughout the semester; in a workplace setting, these firms would be paying clients for the analyses being conducted. This course prepares students to succeed in the workplace, and maintaining a high degree of professionalism is expected.

Course Schedule

Week	Date	Event/Topics	Quizzes/Practice	Assignments
1	08/26	Course Introduction Getting Started Programming Basics	Practice 1	HW 1(Due 08-Sept. at 8pm)
2	09/02	Labor Day: **No Class** Review Programming Basics	Practice 2	
3	09/09	Functions	Quiz 1 Practice 3	HW 2(Due 22-Sept. at 8pm)
4	09/16	Conditionals & Testing	Practice 4	
5	09/23	Loops	Quiz 2 Practice 5	HW 3(Due 06-Oct. at 8pm)
6	09/30	Strings	Practice 6	
7	10/07	Vectors & Sorting	Quiz 3 Practice 7	HW 4(Due 20-Oct. at 8pm)
8	10/14	*Exam 1* : Covers weeks 1 - 6		
9	10/21	Fall Break: **No Class**		
10	10/28	Matrices	Quiz 4 Practice 8	HW 5(Due 10-Nov. at 8pm)
11	11/04	Data Frames	Practice 9	
12	11/11	Data Visualizations	Quiz 5 Practice 10	HW 6(Due 24-Nov. at 8pm)
13	11/18	*Exam 2* : Covers weeks 7 - 11		

Week	Date	Event/Topics	Quizzes/Practice	Assignments
14	11/25	Monte Carlo Methods	Quiz 6Practice 11	HW 7(Due 08-Dec. at 8pm)
15	12/02	R Markdown		
16	12/09	Final Review		
17	12/11	Make up day		
17	12/12	Finals	Date TBD	

References

Some content on this page has been modified from other courses, including:

- CMU 15-112: Fundamentals of Programming, by David Kosbie & Kelly Rivers