INST 630 – Introduction to Programming for the Info Professional
Spring 2019 - Tentative Syllabus V1 (1/22/2019)

This syllabus is subject to change throughout the semester. Check Canvas (ELMS) for the current version.

Personnel:	Vedat G. Diker	Talal Munzar	**Class meeting time and place:**
Office:	Hornbake 0216D	(TA)	Monday
Phone:	(301) 405-9814		*(1/28/2019 - 5/13/2019)*
E-mail:	Use ELMS messages	Use ELMS messages	6:00 PM to 8:45 PM
Office Hours:	By appointment	As announced	in Atlantic Bldg. (ATL) 2416

Catalog Description

An introduction to computer programming intended for students with no previous programming experience. Topics include fundamentals of programming and current trends in user interface implementation that are relevant to information professionals.

Extended Course Description

This course is an introduction to computer programming intended for students with no previous programming experience. Topics include fundamentals of programming, such as variables, data types, assignments, nesting, loops, arrays, functions, objects and storage. The course will also touch on current trends in user interface implementation that are relevant to information professionals.

This course also provides opportunities to develop an understanding of how programming is situated in and reflects broader social structures, constructs and issues, e.g. race, class or gender. Programming is often viewed as a value-neutral technical skill. However, the social and cultural impacts of information and technology are central concepts in our field, and any informed professional needs to understand how these issues manifest in a variety of circumstances. Mostly through readings, but also through discussion and writing, we will critically examine issues of racism, sexism and other forms of bias, inequity and oppression that are pervasive in programming and related technical activities.

Student Learning Outcomes

Upon successful completion of the course, students will be able to:

- Explain basic programming concepts and techniques, and important concepts for the development of interactive web applications.
- Apply concepts and techniques of computer programming, including variables, data types, assignments, loops, arrays, functions, objects, storage, event programming, and toolkits, to create and debug interactive web applications.
- Explain how programming is situated in and reflects broader social structures, constructs and issues, e.g. race, class or gender.
- Articulate strategies and identify resources for ongoing professional development and learning about web programming.

Teaching Notes

This is an introductory course. If you have some experience programming, you will find much of the material similar to what you already know – programming languages tend to be more similar than different at this level. If this is the case and you are interested in being challenged, I invite you to talk to me about leading a session, (it's really true that you learn more by teaching), identifying more challenging exercises or developing a more ambitious project.

Each week will typically follow this pattern, with some exceptions:

Before class (preparation):

- Do assigned readings; watch assigned videos;
- Work on preparation exercises – these help you confirm that you understand the basic material or help you identify specific aspects that you have questions about;
- Complete a low-stakes assignment or quiz before class.

In class:

- We will use a mix of lecture, discussion and hands-on activities to help you apply the materials;
- We will make use of paired and group work in class;
- Class is not a time for solo learning. As members of a learning community, we are mutually responsible to each other as learners. Each of us has to be fully engaged with each other in the activities. We have to be supportive of each other as we try to explain or demonstrate something new, as we inevitably make mistakes. We aren't successful unless everyone is learning.

After class (homework):

- There will be follow-up activities as homework to help you practice, reflect and extend your understandings. You usually have the option of either working together or solo on these, as long as the final work product is your own.

During the second half of the semester, after you have some time to focus on the nuts and bolts of JavaScript, we will start examining broader issues of programming and coding – the social and organizational context, issues related to gender, race, disability, etc. This will help you prepare for situations that you are likely to encounter in your professional work.

To use our time effectively, you must come to class on time and prepared. Being prepared for class means that you have:

a. Completed all assigned readings/videos;
b. Completed all assigned preparation exercises;
c. Either successfully completed them or submitted your questions before class, so I have time to prepare and answer them in class.

We will have an online discussion forum on which you can ask questions and state problems, seeking input from me, the course TA, and other students in the course. You can discuss course-related issues on the forum with the exception of providing direct answers or solutions to graded assessment elements such as assignments, exams, and projects.

Homework assignments are typically due Saturday or Sunday night. I may post and discuss the solutions in class on Monday, so late submissions will not be accepted after class starts and will receive a zero grade.

Here is my suggested general strategy for working on activities:

1. Start early – don't wait. That will give you time to work through the problems and get help as needed.
2. When you run into a problem, spend 5-10 minutes trying to solve it on your own.
3. Then take a break. Sometimes this will allow you to come back and see something you missed. Letting your sub-conscious work on it for a while (unsupervised, so to speak) will often lead to useful ideas.
4. If you've spent 20-30 minutes and still are stuck, post your question the General Discussion forum. We are here to help each other, so don't beat your head against a brick wall - ask for help! When you post, provide as much information as you can. Often it helps to post a screenshot with the problem.
5. As an alternative to the discussion forum, (or in addition to posting on it), send your question or problem to the TA and me via email or Canvas message. Make sure that you attached your code files to your email or Canvas message.
6. If you see a question that you can answer, or if you have an idea, please respond. Don't wait for me. You will be helping your colleagues.

Textbooks and readings

1) Our primary textbook is:

> Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Development.* Indianapolis, Indiana:Wiley. ISBN-13: 978-1118531648.

2) If you don't already know HTML & CSS, you may consider buying the optional textbook *HTML and CSS: Design and Build Websites* by the same author. (Note that the HTML/CSS book is not required). The HTML/CSS book and the Javascript book listed above may be available as a set through some online bookstores with substantial savings.

3) We may also refer to this online, freely available book; consider downloading it at this time (https://eloquentjavascript.net/2nd_edition/):

> Haverbeke, M. (2014). *Eloquent JavaScript, 2nd. Ed.* No Starch Press. ISBN: 978-1-59327-584-6.

4) Other readings and videos may be made available through ELMS.

Required Technology

We will do live programming exercises during most classes, so bring your laptop and be prepared to write code. Any operating system will do. If you don't have access to a laptop, contact me before the first class. The two pieces of software you will need for this course are a web browser, such as Firefox, Chrome, Safari or Edge, and an advanced text or code editor such as BBEdit, Notepad++, Sublime, Brackets, Blue Griffon or Visual Studio Code (VS Code).

Grading

Your final grade for the course is computed as the sum of your scores on the individual elements below (100 possible points total), converted to a letter grade:

A+ 98-100	B+ 87-89.99	C+ 77-79.99	D+ 67-69.99	F 0-59.99
A 93-97.99	B 83-86.99	C 73-76.99	D 63-66.99	
A- 90-92.99	B- 80-82.99	C- 70-72.99	D- 60-62.99	

Graded item	**Percent of final grade**
Preparation – Assignments or low stakes quizzes that you submit before each class.	10%
In class – Reflects your active and visible engagement with the in-class activities.	5%
Homework – Will usually be assigned biweekly, with some exceptions. Includes coding problems (of course), but may also include analysis questions, brief reflective writing, and other activities.	30%
Midterm – This is a diagnostic for you to assess your understanding of the programming basics that are necessary as you move into the project. You will want to address any weaknesses this diagnostic identifies to ensure you are well prepared for the project. This will be in the form a take home assignment.	15%
Project – The project will give you an opportunity to apply and extend what you learn in class. Starting early in the semester, you will work with me to develop projects that match your interests and needs. Given the section size, you will work in pairs as teams. Working as a pair/team allows you to undertake a more ambitious and rewarding project. Project deliverables are the proposal, prototypes, your class presentation, and the final project report.	30%
Critical reflection - At the end of the semester you will submit a critical reflection essay that builds on our critical readings. It will also address the course, your learning, and your plans for professional growth in this area.	10%

A Note on Policies

The essential purpose of the university's policies (https://president.umd.edu/administration/policies) is to enable all of us to fully participate in an equitable, accessible and safe academic environment so that we each can be challenged to learn and contribute most effectively. Policies are, by necessity, often written in impersonal, legalistic language. Nevertheless, we are all responsible for following them. The following sections summarize selected policies as implemented for this course, and provide links to additional information. We are all responsible for knowing and following all university policies.

Academic Integrity

Academic dishonesty is a corrosive force in the academic life of a university. It jeopardizes the quality of education and depreciates the genuine achievements of others. Apathy or acquiescence in the presence of academic dishonesty is not a neutral act. All members of the University Community -

students, faculty, and staff - share the responsibility to challenge and make known acts of apparent academic dishonesty. As a student, you have a responsibility to avoid violations of the Code of Academic Integrity. This includes:

- Cheating: "Intentionally using or attempting to use unauthorized materials, information, or study aids in any academic exercise."
- Fabrication: "Intentional and unauthorized falsification or invention of any information or citation in an academic exercise."
- Facilitating Academic Dishonesty: "Intentionally or knowingly helping or attempting to help another to commit an act of academic dishonesty."
- Plagiarism: "Intentionally or knowingly representing the words or ideas of another as one's own in an academic exercise."

For additional information on the Code of Academic Integrity see shc.umd.edu/SHC/StudentAcademicDishonesty.aspx.

Students with Disabilities

The University is legally obligated to provide appropriate accommodations for students with disabilities. The campus' Accessibility & Disability Service (ADS) works with students and faculty to address a variety of issues ranging from test anxiety to physical and psychological disabilities. If a student or instructor believes that the student may have a disability, they should consult with ADS (301-314-7682, adsfrontdesk@umd.edu, https://www.counseling.umd.edu/ads/). To receive accommodations, students must first have their disabilities documented by ADS. The office then prepares an Accommodation Letter for course instructors regarding needed accommodations. Students are responsible for presenting this letter to their instructors.

Attendance Policy

University policy excuses the absences of students for illness, religious observances, participation in University activities at the request of university authorities and compelling circumstances beyond the student's control. Students who miss a single class for a medical reason are not required to provide medical documentation, but students who are absent more than once are responsible for providing various forms of documentation, depending on the nature of the absence. For additional information on attendance policies, see https://www.usmd.edu/regents/bylaws/SectionIII/III510.html (religious observance) and https://president.umd.edu/administration/policies/section-v-student-affairs/v-100g (medical absence).

Late Work

I do not accept late work except for university-sanctioned excuses that are properly documented (satisfying certain documentation standards). If know you will have to miss a deadline, you must inform me as soon as you know about the matter, indicating the reason and when you propose to submit your work. This includes reasons such as religious observance. If you end up missing a deadline due to an unplanned emergency, such as an illness, you must inform me as soon as possible after the emergency occurs, and provide documentation to substantiate your excuse. Some graded work, such as assignments, will not be accepted at all without a university-sanctioned excuse. I may accept other types of graded work late at my discretion, and the general policy in those cases is that late work will be deducted 20% of its total grade per calendar day, starting on the same day it is due, unless otherwise stated. If you have a legitimate reason, such as a major medical or family emergency, I may agree to an extension or makeup work, which I will grade by the end of the semester. Documentation of the emergency (e.g. a doctor's letter) will be required and must meet certain documentation standards, such as indicating the exact dates on which your doctor suggests you are excused from school work.

Course Evaluation

Course evaluations are a part of the process by which the University of Maryland seeks to improve teaching and learning. The University Senate approved the implementation of a standard, online, University-wide course evaluation instrument. Each course evaluation contains a set of universal questions, and some are supplemented by questions from specific colleges. Students who leave no "Pending" evaluations in their Evaluation Dashboard each semester can view the aggregate results of a sub-set of universal items online. Across the University, course evaluations are being administered through a web-based system called CourseEvalUM. All information submitted to the Evaluation System is confidential. Instructors and academic administrators can only view summarized evaluation results after final grades have been submitted. Instructors and academic administrators cannot identify which submissions belong to which students. This standardized set of evaluation results provides the University with useful information on teaching and student learning across the campus. For additional info see Student Fast Facts at https://www.irpa.umd.edu/Assessment/CourseEval/StuFastFacts.html.

Emergency Preparedness

See: https://prepare.umd.edu/

Syllabus Change Policy

This syllabus is a guide for the course and is subject to change with advance notice. Changes will be posted in ELMS. The ELMS course space is the definitive location for all course work, and communication, including class schedules, assignments and deadlines.

Schedule

This preliminary schedule provides *approximate* dates of topics and major assignments. The ELMS course space will be used to provide specific dates and detailed information on all assignments (major and otherwise). I may have to miss part or whole of some class meetings. While I am away, class sessions may be handled as online, asynchronous weeks, by the course TA, and/or as guest lecture.

Week	**Topics** • **Readings/activities (due before class, except Week 1)**	**Major assignm't**
1. Jan. 28	Introductions, overview of course, structure • Syllabus • Ground rules Computational thinking • Sequence, loops, conditionals – Stage 2, The Maze, from Code.org, https://studio.code.org/s/20-hour	

	HTML and CSS basics • Slides • Selected sections from Introduction to HTML (http://infopeople.org/resources/tutorials/html) Introductory questionnaire • On Canvas	
2. Feb. 4	Computational thinking • Wing, J. (2006). Computational Thinking. *Communications of the ACM.* http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf • Carver, C. (2013). Things I wish someone had told me when I was learning to code. Medium. https://medium.freecodecamp.com/things-i-wish-someone-had-told-me-when-i-was-learning-how-to-code-565fc9dcb329 • Eloquent JavaScript – Introduction (through "On Programming") • More loops, functions – Stages 9 & 11 from Code.org, https://studio.code.org/s/20-hour Decisions and loops • Duckett - Ch 4	
3. Feb. 11	JavaScript and the web browser • Duckett – Introduction & Ch 1 • Optional: Eloquent JavaScript – Ch 12 Computational thinking • Pseudocode – Eloquent JavaScript ("Why Language Matters", pp. 3-6) JavaScript – Statements, variables, values, data types, operators, expressions • Duckett – Ch 2	
4. Feb. 18	Functions, Arrays • Duckett - Ch 3, pp 85-117	
5. Feb. 25	Objects • Duckett -Ch 3, pp 118-144 Document Object Model • Duckett - Ch 5	Project plan
6. Mar. 4	Review Week	
7. Mar. 11	HTML Forms and Form Handling in JavaScript • Duckett - Ch 13	
8. Mar. 18	**Spring Break - No class**	
9. Mar. 25	Event programming	Project

	- Duckett - Ch 6	proposal
10. Apr. 1	Testing and debugging - Eloquent JavaScript - Ch 8 (through “Debugging”, pp. 134-138) - Duckett - selections from Ch 10 Coding conventions and code quality tools. Critical perspective: Meritocracy; Barriers to diversity	
11. Apr. 8	Using JavaScript libraries: JQuery - Duckett - Ch 7 Critical perspective: Sociotechnical systems; Social good	Midterm
12. Apr. 15	Ajax and JSON - Duckett - Ch 8 Critical perspective: Search engine bias; Algorithm bias	
13. Apr. 22	Review Week **(Thanksgiving week)** Critical perspective: Coding culture and diversity Team Project work – Aspired stage: Proto 1	
14. Apr. 29	APIs - Browsers, libraries and platforms - Duckett - Ch 9 Critical perspective: Accessibility, usability, universal design Team Project work – Aspired stage: Proto 2	
15. May 6	Team Project work – Aspired stage: Proto 3 / Wrap-up	
16. May 13	**Last class** Project presentations and discussions Ongoing professional development Course evaluations	Project presentation
17. May 17 (**This is a FRIDAY**)	**Deadline for project report and critical reflection**	Project report and critical reflection