Molloy College

COURSE OUTLINE

DEPARTMENT:	PROFESSOR:	SEMESTER:
Mathematics and Computer Studies	Dr. Jason Schanker	Fall 2017
TITLE: Introduction to Web Scripting	COURSE #: CIS/CSC 235	SECTION: 01

DESCRIPTION:

An introduction to programming through the use of a web scripting language such as JavaScript. Topics include variables, functions, conditional statements, loops, arrays, objects, and event handling. Students will also learn about the Document Object Model and how to manipulate it to make interactive web pages. The course culminates in the creation of a fully functional graphical web application game. 3 credits

MOLLOY COLLEGE
COURSE OUTLINE

Introduction to Web Scripting: CIS/CSC 235
Fall 2017 Semester
Professor: Jason Schanker
Office: 218 Casey (next to the C217 computer lab, C219 is the math/cs office)
Telephone: 516-323-3430 (E-mail checked more frequently)
E-mail Address: jschanker@molloy.edu
Office Hours: Monday - Thursday 6:30-7 p.m. and by appointment

Prerequisites:
None

Course Goal:
To provide students with an introduction to object oriented programming and modern JavaScript while providing experience in the application of these skills to collaborative projects and to the creation of games.

Student Learning Objectives:
By the end of this course, students will be able to:

- Effectively use the fundamental constructs of programming languages: Primitive data types, Objects, Conditionals, Loops, etc.
- Appropriately define classes to encapsulate data and to separate responsibilities (e.g., separating the model from the view)
- Design and implement interfaces
- Create 2D HTML5 Canvas games according to sound object oriented design principles.
- Apply basic math, common game algorithms, and problem solving to the development of games, which includes creating animations, detecting collisions, basic artificial intelligence, etc.
- Utilize basic event-driven programming, callbacks, and higher order functions
- Use JavaScript to modify the structure of the Document Object Model on the fly
- Write programs in a modern style of JavaScript, using some of the major feature additions and syntax standardized in ECMAScript 2015
- Use GitHub to collaborate, post code to a repository, create and merge branches, and pull updates from another repository
- Appropriately document code using JSDoc
- Effectively use initially unfamiliar Application Programming Interfaces by reading documentation and Q & A forums such as StackOverflow
- Decompose large problems into more manageable ones and solve them.
- Create mobile applications using React Native/Expo

Assessment Measures (see Criteria for Grading for percentages):

1. Quizzes
2. Code Reviews/Discussions/Logs
3. In-class/Homework Exercises and Project Work Tasks
4. Final Project/Oral Presentation

Criteria for Grading:

1. Quizzes: **10%**
   There will be short quizzes assessing your understanding of the content covered in lecture and your ability to apply what you learned through completing in-class/homework exercises and project work tasks. Make-up quizzes will be given at my discretion.

2. Code Reviews/Discussions/Logs, part of Molloy College's Communicating Across the Curriculum: **10%**

   For this portion of your grade, you will be participating in discussions on Canvas and GitHub, which include identifying bugs or inefficiencies in other students' code, asking clarifying questions, and coordination of the division of labor for the collaborative final project. Discussions should also be used to ask questions on concepts or exercises and to provide answers to your peers' questions. When you complete weekly project tasks, you will also submit short weekly logs, which reflect on what was most challenging, what you learned, what you found to be the most interesting, the hardest error/bug you encountered and your method of overcoming it, what you enjoyed the most, and any disagreements you may have had when others reviewed your code. These will all be very important talking points for software engineering interviews. (See page 40 of Cracking the Coding Interview: 150 Programming Questions by Gayle Laakmann.) You should strive for technical correctness and clarity and use grammatically correct sentences to practice for formal correspondence.

3. In-class/Homework Exercises and Project Work Tasks: **60%**
   This is a project-based class and as such, most of the time for this class will be spent applying what's taught in class to the creation of web/mobile apps such as Flappy Bird, the Battle of the Birds (e.g., creating the ultimate autonomous bird for competition with other students' birds), Arkanoid, and a collaborative work. You will be developing components of the individual games on a weekly basis and you will be expected to create a schedule with your group project, subject to my approval. Part of the grade for this category is determined by the extent to which you meet the periodic due dates. Note, that it is extremely common (in industry as well) for work to take much longer than expected, particularly when you first start programming, so you should plan for this in your schedule by allowing extra time (at least twice what you expect). As outlined in Grading Criteria 1, these tasks will include a written reflection of your work. You will need to sign up for an account on GitHub so you can post all of your code and publish your work. To learn the basics of GitHub, you may want to follow the quick tutorial at https://guides.github.com/activities/helloworld/. There will also be exercises given in class, and you will be given time to work on them as well as on project components. Exercises/project work that is not finished in class will be assigned for homework. Since much of the content of the course and the project tasks build on itself, it is very important that you don't fall behind! **Be sure to work on exercises/tasks daily to make sure you can ask me questions and complete them by the due date.**

   DISCUSSION/COLLABORATION: As mentioned in Grading Criteria 2, part of your grade will consist of the quality and frequency of comments made on your fellow students' code and answers to their questions when they get stuck. However, **do NOT answer questions by providing your own solutions.** Instead, help by providing hints that get the other student to arrive at his/her own answer. This can help you be a better programmer as you will potentially think of alternate ways of solving the problem and should better prepare the student you're assisting for in-class exams. You should ask for help only after you've thoroughly tried the problem yourself. Try to avoid the discussions until you've solved the problem or have exhausted your brain after multiple attempts. Remember frustration is part of the process!

   > From the NYC Tech Talent Pipeline Report: "As part of this process, candidates must be able to demonstrate their ability to overcome frustration effectively. When one approach doesn't work, rather than becoming despondent or giving up, strong candidates are able to take a step back, regroup, and try a different way."

4. Final Project/Oral Presentation: **20%**
   Throughout the semester, you will be collaboratively developing a web/mobile application that applies what you've learned and adheres to the principles of sound software engineering taught throughout the course. Students will be able to choose what they want to develop, subject to my approval. Ideally, the web/mobile app will be something that you're passionate about developing and would want to spend lots of time on it. One or more groups will be formed according to the chosen project (e.g., all students wanting to recreate a portion of the NES Zelda game would be in one group). An effective collaboration will require an equitable division of responsibilities, strong team management and planning, effective use of version control, clean, organized, and well-documented code, and periodic code reviews. Your grade for the web/mobile application will be based on the extent to which you met the minimum requirements, the content, the soundness of the design (both of the code and the style), the quality of your contributions, your group's design process, and what you learned along the way; it does not need to be fully implemented, although it should be usable in some way. The final group project will be due on the final day of class (12/13 or 12/18). Each team member will also submit an evaluation of their own work as well as the work of their collaborators. Note: Members from the same team may receive different grades.

   At the course's conclusion (12/13 or 12/18), each group will deliver an oral presentation in which each team member discusses his/her contributions and how they fit into the project as a whole. You will need to describe the development process and justify your design decisions, mentioning possible alternatives that were considered and the rationale for selecting some alternatives over others. Team members will also discuss their group’s workflow and what collaboration techniques were or were not effective. Obstacles that were not overcome as well as what was tried and what might work to overcome them should be discussed. Possibilities for future additions and/or improvements should be mentioned as well. Each member’s final presentation grade will be determined by how well the individual communicates an understanding of his/her own contributions (technical correctness and clarity), the value of each individual’s contribution to the presentation as a whole, the extent to which the group addresses the points mentioned above, and the presentation’s overall effectiveness and organization.

Outline of Topics - Tentative Class Schedule

Session	Dates	Topic
Week 1 (1)	**closed**, 9/6	**9/4: Labor Day** Using GitHub HTML, CSS, JavaScript, Output, Variables, Functions, & Creating Animations w/Canvas
Week 2 (2, 3)	9/11, 9/13	HTML, CSS, JavaScript, Output, Variables, Functions, & Creating Animations w/Canvas (cont.) Introduction to Classes and Object Oriented Programming, JSDoc, Pseudorandom Number Generators (PRNGs)
Week 3 (4, 5)	9/18, 9/20	Classes and Object Oriented Programming (cont.) Scaling for screens with different sizes and resolutions Event Handling, Implementing Model/View/Controller classes for Flappy Bird
Week 4 (6, 7)	9/25, 9/27	Arrays, Event Handling, Implementing Model/View/Controller classes for Flappy Bird (cont.)
Week 5 (8, 9)	10/2, 10/4	Using sprite sheets for Animations, Making Decisions, Loops, Collision Detection
Week 6 (10)	**closed**, 10/11	**10/9: Columbus Day** Finishing Touches on Flappy Bird, customizing for Mobile, Using React Native/Expo to make Flappy Bird a mobile app
Week 7 (11, 12)	10/16, 10/18	Implementing Interfaces, Composition/Aggregation, Polymorphism, Battle of the Birds
Week 8 (13, 14)	10/23, 10/25	Battle of the Birds (cont.)
Week 9 (15, 16)	10/30, 11/1	Nested Loops, Creating a Brick Grid For Arkanoid Level, Work on Arkanoid
Week 10 (17, 18)	11/6, 11/8	Inheritance, Arkanoid (cont.)
Week 11 (19, 20)	11/13, 11/15	Arkanoid (cont.), Introduction to Babylon.js and creating 3D Arkanoid levels
Week 12 (21)	11/20, closed	Finishing Touches on Arkanoid **11/22: Thanksgiving Recess**
Week 13 (22, 23)	11/27, **11/29**	Final Projects, using APIs
Week 14 (24, 25)	12/4, 12/6	Final Projects (cont.)
Week 15 (26)	12/11	Finishing Touches on Final Projects
Week 16 (FINAL CLASS)	**12/13 or 12/18**	**Final Projects due, Final Presentations**

Textbook References:

1. Beginner Level: Head First HTML and CSS, 2nd edition by Robson and Freeman, 2012, ISBN-13: 9780596159900
2. Beginner Level: Head First JavaScript Programming by Eric T. Freeman and Elisabeth Robson, O'Reilly Media, 1st edition, 2014, ISBN13: 978-1449340131.
3. Intermediate Level (Available as Free Online Text): Eloquent JavaScript: A Modern Introduction to Programming by Marijn Haverbeke, No Starch Press, 2nd edition, 2014, ISBN13: 978-1593275846. (available at http://eloquentjavascript.net): A comprehensive free online book for learning JavaScript (includes the basics of Node.js)
4. Intermediate Level: Professional JavaScript for Web Developers by Nicholas C. Zakas,Wrox, 3rd edition, 2012, ISBN-13: 978-1118026694
5. Intermediate Level (Available Online): Understanding ECMAScript 6 by Nicholas C. Zakas ("Content for the ebook" available at: https://github.com/nzakas/understandinges6/tree/master/manuscript): Comprehensive book on the new additions to JavaScript

Some Additional Resources:

1. GitHub: For saving your code (so-called code repository hosting), tracking changes made to it, working collaboratively on projects, and publishing your projects for the world to see
2. JSDoc: For specifying what your code does so other people know how to use it (used to generate so-called API documentation)
3. Mozilla Developer Network (MDN) Web Docs: Use as a reference for utilizing/understanding HTML/CSS/JavaScript code
4. React Native: Documentation for developing mobile applications using JavaScript
5. repl.it React Native: Allows you to quickly create/test apps on your mobile device using Expo/React Native
6. Babylon.js: So called 3D game engine (partial description from the code repository on GitHub: "a complete JavaScript framework for building 3D games")

Attendance Policy:
It is important that you attend all classes and complete all in-class exercises, homework assignments, and project work tasks on time. You are responsible for all material covered in class and on the homework. If you have to miss a class, it is your responsibility to get the material for that class and make up any missed work. Exercises and Project Work Tasks, which will be partially completed in class and partially at home, account for 60% of your grade (see above Grading Criteria 3) so missing class can hurt this portion of your average or require you to do more homework. Additionally work on the collaborative Final Project will be done in class so missing class can affect this portion of your average as well (See Grading Criteria 4). Additionally, as mentioned above, if you miss part or all of a class, you quiz average may be adversely affected (see above Grading Criteria 1). Please see the Student Handbook for College Policy regarding Excused Absences.
You are responsible for taking quizzes at the specified time. If you have a documented reason for missing a quiz, contact me prior to the assessment. Make-up quizzes will be given at my discretion.
Be sure to monitor your Molloy email on a regular basis; most communications outside of class will be by email.
It is the accepted practice at Molloy College that faculty take attendance in all courses.
- Students should notify faculty if an absence if necessary as the result of a serious situation.
- Failure to attend class for two (2) consecutive weeks at any point in the semester, without notification of extenuating circumstances, will result in an administrative withdrawal from the course.
- Administrative withdrawal results in removal from the course with a grade of "WA" or "WF" determined by the point in the term and the academic performance.
- Students should consult the College catalog for complete details regarding withdrawals and the potential financial implications of a withdrawal.

Statement on Academic Integrity:
All students are bound by the Molloy College Academic Integrity Policy included in the Student Handbook. Please read it and abide by it. In a class with a forced curve (this is **NOT** such a course) in which there is a fixed percentage of As, Bs, and Cs, a person who commits an act of academic dishonesty and receives a higher grade causes another student to receive a lower grade. If a person gets an A in a class through academic dishonesty and lacks the skills an employer would expect after receiving such a grade in that course, he/she will be unable to get the job. In addition, the employer may not interview other candidates legitimately receiving an A in that class because they may question if the grade represents mastery of the material. Even committing academic dishonesty to "pass" a class can devalue the degree as a certain level of competency would still be expected. This is **NOT** hypothetical; there are instances where employers have mistrusted the value of a degree because of their negative experiences with graduates from the school from which it was earned. **In short, academic dishonesty hurts your fellow students and is therefore unacceptable in every instance.**

Statement on Communicating Across the Curriculum:
Molloy College's Communicating Across the Curriculum requirements are satisfied by Grading Criteria 2: Code Reviews, Discussions, and Logs, accounting for 10% of the final grade. The assignments from this category require students to coordinate project tasks and read other people's code, asking questions and making comments on code that is difficult to understand, doesn't work as intended, or is inefficient in some way. Discussions will also include questions on concepts or exercises and answers from peers. Logs will consist of written reflections of one's own experiences when working on projects. The requirement is also satisfied by the oral presentation of Grading Criteria 4: Final Project/Oral Presentation in which part of the student's grade will be determined by the presentation's clarity, technical correctness, and effectiveness. Finally, students will be writing and documenting code.

Statement on Disabilities:
Students with documented disabilities who believe they may need accommodations in this class are encouraged to contact the Director of the Disabilities Support Service Office, Casey Building, Room 11. The telephone number is 516-323-3315.