Introduction to Programming

Course Syllabus 2020-2021

Instructor:	Aaron Spike, spikeac@mlc-wels.edu
Grade Level:	9 - 12
Course Length:	1 semester
Credit:	0.5 credit
Prerequisites:	none

Description:

Introduction to Programming is a hands-on introductory computer-programming course. Students will learn to write computer programs using the ubiquitous JavaScript computer programming language. In this process, students will exercise and grow their God given technology skills and prepare to use these gifts in service in their homes, churches, communities, and future vocations.

Course Outcomes:

Upon successful completion of this course, students will:

- Operate a text editor and browser base developer tools
- Locate and learn from online programming documentation
- Be acquainted with the main features of structured programming languages
- Compose syntactically correct JavaScript programs
- Recognize and reproduce some common programming idioms
- Design simple web-based user interfaces
- Read and talk about code
- View programming as a powerful, but limited, tool for modeling and simulating systems
- Utilize and appreciate the computer and computer programming as a tool for solving problems and simplifying tasks

Course Topics:

- A Programmer’s tools
- Mathematical expressions
- Variables
- Basic HTML and CSS
- Functions and events
- Boolean expressions
- Conditional statements
- Strings
- Common data structures
- Loops
- Objects
- Using Browser APIs
- Using 3rd-party Web APIs
- Using Libraries

Course Materials:

No textbook is required for the course.

Our course site links to freely available, online documentation from a variety of websites including YouTube.com.

Students will need access to a computer or Chromebook with internet access and the following software tools:

- The most recent version of the Google Chrome web browser - https://www.google.com/chrome/
- The Caret text editor extension - https://chrome.google.com/webstore/detail/caret/fljalecfjciodhpcledpamjachpmelml?hl=en
- The Web Server for Chrome extension - https://chrome.google.com/webstore/detail/web-server-for-chrome/ofhbbkphhbklhfoeikjpcbhemlocgigb?hl=en

Evaluation & Grading Scale:

Assignments: Each graded assignment in the course is assigned a certain number of points. The course grade will be determined by comparing the points earned to points possible and then converting this percentage to a letter grade based on the grading scale below. Points may be deducted from late assignments. A student's current grade in the course will be available as we progress and should be checked regularly to track progress.

Grading Scale:

98-100 A+	87-89 B+	87-89 C+	67-69 D+	
93-97 A	83-86 B	73-76 C	63-66 D	0-59 F
90-92 A-	80-82 B-	70+72 C-	60-62 D-	

Fellow programmers reading this syllabus will undoubtedly notice mathematical ambiguities in the grading scale specified above. What letter grade do percentages between 89 and 90 merit? Translating ambiguous specifications into correct code is a daily struggle for the working programmer and therefore each student in this class. In the interest of clarity and completeness, I offer the following code sample for calculating course grades. By the end of the course, you will know how to read it.

```
const calculate_final_grade = function(points_earned_by_student, points_possible_in_course){
  const percentage = points_earned_by_student / points_possible_in_course;

  let letter_grade;

  if (percentage >= 98) {
    letter_grade = 'A+';
  } else if (percentage >= 93) {
    letter_grade = 'A';
  } else if (percentage >= 90) {
    letter_grade = 'A-';
  } else if (percentage >= 87) {
    letter_grade = 'B+';
  } else if (percentage >= 83) {
    letter_grade = 'B';
  } else if (percentage >= 80) {
    letter_grade = 'B-';
  } else if (percentage >= 77) {
    letter_grade = 'C+';
  } else if (percentage >= 73) {
    letter_grade = 'C';
  } else if (percentage >= 70) {
    letter_grade = 'C-';
  } else if (percentage >= 67) {
    letter_grade = 'D+';
  } else if (percentage >= 63) {
    letter_grade = 'D';
  } else if (percentage >= 60) {
    letter_grade = 'D-';
  } else {
    letter_grade = 'F';
  }

  return letter_grade;
};
```

Academic Honesty

As the Latin proverb says,”Repetitio mater studiorum est.” Repetition is the mother of learning. Students in *Introduction to Programming* will build fundamental computer programming skills through regular practice. In the process, students are encouraged to discuss, collaborate, and assist each other. But this is very much a learning-by-doing class. No one should miss out on the benefit that comes from completing the assignments. It is imperative that each student complete every assignment from start to finish on their own computer with their own hands. Students must practice the skills so that they will not be caught off guard on the final exam where collaboration will not be allowed.

Because collaboration and individual work often seem at odds, let's make an analogy to math class to help clarify what is and is not acceptable behavior:

Scenario #1: Two students are working on their math homework together. One student writes all of the work and answers down on a piece of paper and makes a photocopy. The first student turns in the original and the second turns in the photocopy.

Scenario #2: Two students are working on their math homework together. As they discuss the problems, each student writes all of the work and answers down on their own piece of paper in their own handwriting. Each student turns in their own paper.

In Scenario #1 one student lost the opportunity to practice organizing their thoughts on paper and writing mathematical notation. Even though both students worked on the assignment, Scenario #1 looks an awful lot like cheating. The students in Scenario #2 may have had the same discussions as they completed the work, but because they both wrote their own copy, they both derive benefit from the physical practice and it arouses no suspicion.

In *Intro to Programming*, it is OK for students to sit right next to each other and build their programs together, each person on their own computer, even if the results are very similar. But it is not OK for multiple students to sit in front of one computer, build one program, and hand in two copies of the same file. Please collaborate, discuss, interpret the directions together, review each other's work. DO NOT hand in the same work.

If you need additional clarification or have questions about whether or not I might consider something cheating, please ask!

Instructor Policies:

Expected of Students:

1. Check in on the course on a daily basis. Generally, students have been scheduled by their school to work on this course at a specific time each school day. Be faithful in using this time to work on this course. If you will be absent from class on a particular day, or if you miss class because of illness, be sure to let me know.
2. Expectations for this course are high, but reasonable. Everyone is very busy with their other courses, activities, and responsibilities, yet you are expected to complete all work on time. If you need an extension of time, please make the request via private email prior to the due date.
3. Some assignments and activities may have due dates on weekends or during days when your school is off. This is done to give you extra work time and to help you balance the workload with your other classes. If you are unable or prefer not to work at those times, please adjust your schedule and pace to complete the work early.
4. Communicate publically. Much of a working programmer's learning happens through public mailing lists and forums. All questions of technical nature should be asked in the forums on our course site so that your peers can benefit from the exchange. Private email should be reserved for questions of a personal nature, e.g. excuses and grading questions. Good two-way communication is very important for a successful online learning experience!

ALHS Online Policies:

Current ALHS Online policies are listed in the ***Handbook for ALHS Online Students and Parents***, available on the ALHSO.org website. This includes policies on non-discrimination, anti-harassment, student expectations, attendance, academic honesty, student discipline, student grades, course add/drop, etc.

Please note the policy on **class attendance** which states in part:

> ***"Even if a student's local school does not have school on a particular day (snow day, teacher's conference, quarter break, choral fest, class trip, etc.) ALHS Online courses will continue to meet and students are expected to complete required work on time."***

Students also fall under the policies of the school where they attend as a full-time student. When applicable, these same local school policies will be applied to enrollment in this ALHSO course.