MVLA
2025-26
COURSE INFORMATION SHEET

Course Title: ADEN
CTE Pathway Sequence: Information and Communication Technologies – Software and Systems Development
School: Los Altos High School
UC/CSU requirement: No (General Elective)
MVLA Graduation requirement: No (General Elective)
Textbook and/or other learning resources:
Introduction to Programming Using Java, *9.0 JavaFX Edition* by David Eck

Course Description/Student Learning Outcomes:

ADEN is a course where students will expand their programming knowledge beyond AP Computer Science, exploring the application of advanced data structures and programming concepts to solve problems. Students will experience all aspects of the system development process through individual and collaborative projects from design specification through programming, testing and debugging, and benchmarking.

Student Learning Outcomes:

By the end of the course, students will be able to:

1. Explore advanced data structures and concepts **and apply them to solving problems.** Specifically, they will be familiar with Stacks and Queues, Trees and Tries, Sets and Map, Hashing and Graphs.
2. Consider the overall performance of multiple solutions and implement an appropriate strategy
3. Develop collaborative programming skills, including use of Software Change Management and regression testing.
4. Interface with real hardware in embedded systems, including I/O, analog sensors, motors, timers, interrupts and Bluetooth networks.

Course Outline/Units of Study/CTE Industry Standards(If applicable to your course):

ADEN prepares students planning to study computer science or engineering for college and for a career in the California Career Technical Education (CTE) Software and Systems Development pathway. This class covers portions of the following standards of the Software and Systems Development pathway :

CTE.ICT.C.1.0 – Identify and apply the systems development process. (1.1, 1.3 – 1.6)
CTE.ICT.C.2.0 – Define and analyze systems and software requirements. (2.2 – 2.4)
CTE.ICT.C.3.0 – Create effective interfaces between humans and technology. (3.1, 3.2)
CTE.ICT.C.4.0 – Develop software using programming languages (4.1, 4.3 – 4.11)
CTE.ICT.C.5.0 – Test, debug, and improve software development work. (5.1, 5.3 – 5.6)

CTE.ICT.C.6.0 – Integrate a variety of media into development projects. (6.4, 6.6)

CTE.ICT.C.8.0 – Develop databases (8.6, 8.6)
CTE.ICT.C.9.0 – Develop software for a variety of devices, including robotics. (9.1 – 9.5)

This course will cover the following topics, each with one or more programming assignments:

- Introduction: Software Change Management, Java vs C, Java FX and File IO
- Stacks and Queues
- Trees and Tries
- Hashes
- Graphs and Search Algorithms
- Embedded Systems Programming (introduced throughout the year)

Assessment and Grading (BP 5121 / AR 5121**):** To ensure that every student has an equal opportunity to demonstrate their learning, the course instructors implement aligned grading practices and common assessments with the same frequency.

1. Grading categories and their percentage weights:

Programming Assignments and Quizzes	85%
Homework/Classwork/Portfolio	15%

2. Achievement evidence collected within each grading category:

   Daily homework will be assigned and checked. Quizzes will be given during some units as a check for understanding. Every unit will have *at least one* programming performance task and **may** also include a unit assessment. Programming tasks are primarily graded on functionality – however, up to 20% of the grade is based upon adherence to programming guidelines and implementation/architecture of the solution.

3. Grading scales:
   A: 90 – 100%
   B: 80 – 89%
   C: 70 – 79%
   D: 60 – 69%
   F: < 60%

4. Homework/outside of class practices (AR 6154):

   Homework will be assigned on the Canvas class website daily and will be reviewed in the next class. All homework details are listed on the Canvas website. One or two larger programming tasks will be assigned per unit.

   A larger project (or extension of an existing project) may be assigned as the Final project. This will not be called out as a separate grade category; rather, the project itself will be assigned more weight within the Programming Assignments and Quizzes category.

5. Excused absence make-up practices (Education Code 48205(b)):

   Homework will be accepted for full credit with one day allowed for each day of the excused absence. However, students will also be required to keep up with current class assignments. Homework can be submitted multiple times before the due date.

   Per MVLA AR 5121: “Make-up work and late assignments are accepted within a reasonable time frame established at the course level to ensure students' equal educational opportunity.”
   The following absences do not require the instructor to provide make-up work:
   - Unverified Unexcused (U)
   - Verified Unexcused (X)
   - Tardy (T)

- Tardy Truant (Y)

However, students with excessive absences will be supported to the extent possible to limit the impact of absences on the student's grades.

6. Academic integrity violation practices (LAHS Academic Integrity Policy / MVHS Academic Integrity Policy):

   Students are expected to produce their own original work for all assignments, unless otherwise noted (e.g., a classwork assignment on a new topic where students are explicitly told to work together, or a pair programming assignment).

   Most of the learning in computer science comes from figuring out how to do something on one's own. While it is fine to discuss concepts and broad strategies or to ask others for general help on a concept, students should avoid working together to write their programs line-by-line.

   If students receive help from friends, parents, through internet searches or other methods, they should be sure that it is about the general concept being applied, and not specific line-by-line help. If students are asked about certain portions of their program and ***are not able to explain their own work***, this could be considered an academic integrity violation and result in a 0% for the assignment (including for tests and the final project).

   For homework assignments, quizzes and **ALL** programming assignments, code that is found to be copied from another source (e.g., the Internet, another student, etc.) or created using generative AI will receive a 0%.

7. Late work practices:

   Late homework that is late by one to three days may be submitted for a minimum of 80% credit. Homework that is more than three days late may be submitted for a minimum of 60% credit. Missing work will be given a 0%.

   *Note: I **will** consider extension requests for projects **if and only if** the request is made at least 24 hours in advance **and** the student is taking advantage of ACT or office hours to get help.*

8. Revision practices:

   Quizzes: Quizzes cannot be retaken or revised

   Programs: Because programs are graded primarily on functionality **and** I provide students with tests to validate their work before they turn it in, I generally do not regrade programs, although students may choose to accept a late penalty in order to fully debug their code. Students are also supplied with a code style analyzer - so errors in style are also not revisable.

9. Extra credit practices: No extra credit will be given

10. Additional grading practices: None

11. LMS Used: Canvas

Instructors' email addresses:

scott.murray@mvla.net

Additional information:

Materials: Students should bring their own device for use in class. Chromebooks will not be sufficient for the assignments in this class. There are a limited number of laptops available for student use during class.

This class is taught by Career Technical Education certified teachers in Information Technology.