MVLA
2025-2026
COURSE INFORMATION SHEET

Course Title: AP Computer Science
School: Los Altos High School
UC/CSU requirement: Yes/Yes
MVLA Graduation requirement: No
Textbook and/or other learning resources: *Introduction to Programming Using Java*, Ninth Edition by David Eck (http://math.hws.edu/javanotes/)

Course Description/Student Learning Outcomes:Student Learning Outcomes:

AP Computer Science is a rigorous college-level course focused on computer science concepts and Java programming. Upon completion of the course, students will be prepared for the Advanced Placement Computer Science A examination. The course outline follows the syllabus prescribed by the College Board.

Student Learning Outcomes: AP College Board has reorganized the curriculum into four main units: Using Objects and Methods, Selection and Iteration, Class Creation and Data Collections. Students will be able to:

1. Determine an appropriate program design and develop algorithms
2. Write and implement program code
3. Determine the output or result of a given program code segment or explain why code may not work as intended
4. Describe the behavior and conditions that produce specified results in a program
5. Understand the ethical and social implications of computer use

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

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.
CTE.ICT.C.1.3 Identify and describe how specifications and requirements are developed for new and existing software applications.
CTE.ICT.C.1.6 Diagram processes using flowcharts and the Unified Modeling Language.
CTE.ICT.C.2.2 Recognize and prevent unintended consequences of development work: programming errors, security issues, health and environmental risks, and privacy concerns.
CTE.ICT.C.3.1 Describe and apply the basic process of input, processing, and output.
CTE.ICT.C.4.0 Develop software using programming languages.
CTE.ICT.C.4.1 Identify and describe the abstraction level of programming languages from low-level, hardware-based languages to high-level, interpreted, Web-based languages.
CTE.ICT.C.4.10 Create and know the comparative advantages of various queue, sorting, and searching algorithms.
CTE.ICT.C.4.3 Identify and use different authoring tools and integrated development environments (IDEs).
CTE.ICT.C.4.4 Identify and apply data types and encoding.
CTE.ICT.C.4.5 Demonstrate awareness of various programming paradigms, including procedural, object oriented, event-driven, and multithreaded programming.
CTE.ICT.C.4.6 Use proper programming language syntax.
CTE.ICT.C.4.7 Use various data structures, arrays, objects, files, and databases.
CTE.ICT.C.4.8 Use object oriented programming concepts, properties, methods, and inheritance.
CTE.ICT.C.4.9 Create programs using control structures, procedures, functions, parameters, variables, error recovery, and recursion.

CTE.ICT.C.5.0 Test, debug, and improve software development work.
CTE.ICT.C.5.3 Use strategies to optimize code for improved performance.
CTE.ICT.C.5.4 Test software and projects.
CTE.ICT.C.5.5 Evaluate results against initial requirements.
CTE.ICT.C.5.6 Debug software as part of the quality assurance process.

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:
   Homework/Classwork: 15%
   Quizzes/Tests/Summative (Web-CAT) Programming Assignments: 65%
   Final: 20%

2. Achievement evidence collected within each grading category:
   Daily homework will be assigned and checked. Quizzes will be given during some chapters as a check for understanding. Every chapter will have a unit test at the completion of the unit. Chapter tests often involve a programming performance task.

   At the completion of each semester, two or three weeks will be set aside for a final project that encompasses many of the concepts learned in that semester. A multiple-choice final exam may also be given at the end of the semester.

3. Grading scales:
   A 90 - 100% **B** 80 - 90% **C** 70 - 80% **D** 60 - 70% **F** Below 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 (Web-CAT) will be assigned per chapter.

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

6. Academic integrity violation practices (LAHS 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, etc., 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 a 0% for the assignment (including for tests and the final project).

   For homework assignments, quizzes, tests, and WebCAT programming assignments, code that is found to be copied from another source (e.g. the Internet, another student, etc.) will receive a 0%.
   Unless permitted, code submitted that was created using generative AI will receive a 0%.

7. Late work practices:
   Late homework that is late by one day may be submitted for 75% credit. Homework that is more than one day late may be submitted for 50% credit, up until the unit test for the current chapter. Missing work will be given a 0%.

8. Revision practices:
   Quizzes: Quizzes cannot be retaken or revised.

   Tests: Students with a score below 75% will be allowed one retake for that chapter test after seeing the teacher and performing remediation. The retake must be completed before the next chapter assessment. The maximum score allowed for retakes will be 75%.

9. Extra credit practices:
   No extra credit on any assignments will be given.

10. Additional grading practices:
    Students' progress/grades will be updated on the Student Information System at the end of each chapter (typically every two weeks).

Instructors' email addresses:

scott.murray@mvla.net
brent.smith@mvla.net

Additional information:

Materials: Students may bring their own device, or check out a laptop from the laptop cart for use in class. Chromebooks will not be sufficient for the assignments in this class.

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