

AP Computer Science

2016-2017

Room 605

Damien High
School

Mr. Eric McElrea



Alan Turing depicted decoding Nazi messages with the help of his Bletchley Park team.

COURSE DESCRIPTION

The AP Computer Science A course is an introductory course in computer science. Because the design and implementation of computer programs to solve problems involve skills that are fundamental to the study of computer science, a large part of the course is built around the development of computer programs that correctly solve a given problem. These programs should be understandable,

adaptable, and, when appropriate, reusable. At the same time, the design and implementation of computer programs is used as a context for introducing other important aspects of computer science, including the development and analysis of algorithms, the development and use of fundamental data structures, the study of standard algorithms and typical applications, and the use of logic

and formal methods. In addition, the responsible use of these systems is an integral part of the course.

“What’s in your hands I think and hope is intelligence: the ability to see the machine as more than when you were first led up to it that you can make it more.”

- Alan J. Perlis

COURSE RESOURCES

Required text: Paul Deitel and Harvey Deitel , *Java How To Program*, Prentice Hall

Optional text: Roselyn Teukolsky, *Barron’s AP Computer Science A 7th Edition*, Barron’s Educational Series

Course website: moodle.damien-hs.edu

<h2>HOW TO BE SUCCESSFUL IN THIS COURSE</h2> <p>This course, in all likelihood, is your first glimpse into the world of software programming. Even those with some programming experience have probably not been as rigorously involved in the art as this course requires. Here are a few helpful hints to start you on your way to success.</p>	ASSIDUOUS	EXPERIMENTAL	WILLING TO FAIL
	<p>Assiduous is defined as: constant in application or effort; working diligently at a task; persevering; industrious; attentive. Programming requires an attention to detail that you may not be accustomed to. It requires you to constantly rethink and improve your solution. Be ready and open to learn a completely new subject, and do not become frustrated with the hurdles you will face. Persistence and patience is a must.</p>	<p>The best programming students are always experimenting with code on their own. Complete that class project early? Add new features and functionality to it, even if in the end you couldn't get any of those awesome additions working. In the end I promise you that programming and practicing on your own is your best path to success!</p>	<p>To program is to fail. No one gets their code right on the first try (or second, or third, ...). Programming is a process that includes failure after failure after failure. Hopefully each failure gets you gradually closer to a complete and working program, but know now that you will fail many times. We all will. Failure is an important process to learning.</p>

COURSE REQUIREMENTS

This course involves reading, analyzing and writing code. It's comprised of classwork and homework, programming projects, quizzes, tests, and final exam. You will need to put in consistent effort throughout the year to be successful. You'll need to be ready and willing to research and learn on your own. Finally, you will need to demonstrate active learning, not passive absorption.

Learning is not a spectator sport.

Classwork and Homework

(~10-15%) Classwork and homework are comprised of small, one to two day assignments. They represent a

comparatively small portion of your grade.

Programming Projects

(~20-25%) Programming projects are extensive, multiple-day projects that will require you to plan and implement a solution to a given problem. There will be both in-class and take home programming projects.

Quizzes

(~10-15%) Short weekly assessments. 10-15 minute time investment.

Tests

(~20-25%) Unit assessment that will require you to demonstrate your knowledge of lectures,

classwork, homework and programming projects on a unit.

Final Exam

(~20%) Comprehensive exam at the end of each semester that will require you to demonstrate your knowledge on the entire breadth and depth of each unit.



Steve Jobs and Steve Wozniak working at Apple Computers in the early days. (circa 1980)

“It’s fine to celebrate success, but it is more important to heed the lessons of failure.”

- Bill Gates



THE FINE PRINT

Late Work

After an absence, a student is allowed a minimum makeup time, in school days, equivalent to the number of school days missed during the absence, to complete any assignments missing for full credit. A student may also turn in any assignment late for half credit as long as the late assignment is turned in before the corresponding unit test.

Regarding Plagiarism

Programming, often times, requires research and collaboration with peers. You must be conscious and careful not to plagiarize on your programming projects. For the purpose of this class, plagiarism will be defined as taking large amounts of code directly from an online source or a peer. The

best way to learn and become the best programmer you can be, is to put in the work to complete your own projects. However, collaboration or seeking peer help will happen. I ask that you be careful when both helping and receiving help; do not simply give/take the answer or algorithm. Help or receive help on your own algorithm to avoid code duplication that may lead to plagiarism.

Attendance

Being on time and present every day is vitally important. This is a new subject area for you and missing a day or two can prove to be seriously detrimental to your progress and understanding. Please see me in advance if you know you will be absent.

PROGRAMMING PROJECTS

Programming projects are perhaps the most important part of your learning. It is important that you take them seriously and commit to the learning process. Here are a couple of guidelines to keep in mind when it comes to programming projects.

- Projects must be turned in digitally via email, as well as printed and turned in on the assigned due date.
- Late projects will receive a 20% penalty but may be turned in any time after the assigned due date.
- Printer issues are not valid excuses for not having your projects printed. Use the DRC before the due date.

A	B	C	D	F
90-100%	80-90%	70-80%	60-70%	< 60%