2014 AP Computer Science A Syllabus

Michael Ferraro

Balboa High School, Rooms 127 & 319

1 Welcome to APCS

Welcome to AP Computer Science at Balboa High School! This year marks the 7th year APCS is being offered here and I hope you share in my excitement and enthusiasm as we embark on a year-long deep dive into software development using the Java programming language.

While not everyone taking this course will be going on to study programming or Computer Science later, everyone will benefit from the emphasis on problem-solving techniques; abstract thinking and representation; and using computers in ways that casual users almost never do. You will become adept at *Computational Thinking*. By the end of this course, you will feel empowered to use computers to perform a wide range of functions, many of which are applicable to a myriad of professional and academic pursuits.

2 Course Description

AP Computer Science is an approximate equivalent to a one- to two-semester, university-level introductory computer science curriculum. By taking this class, students will learn to

- design, implement, debug, and document computer programs;

- write programs using the Java programming language, an industry standard;

- design and implement modular software components that can be tested, integrated with others, and reused;

- represent information in an object-oriented manner; and

- read and understand APIs[1], which enable you to learn how to use other language features on your own later.

3 Unit Outline

The list that follows is an approximate ordering of topics covered in this course.

1. Basics of Computing

2. Object-Oriented Programming

3. Java Style and Syntax

4. Algorithms and Basic Flow Control

5. Data Types, Variables, and Arithmetic

6. Flow Control via Conditionals

7. More Iterative Statements

8. `FactorQuadratic` Project

9. A Second Look at OO Programming

10. Working with `String`s

---

[1]API stands for *Application Programming Interface*

11. Class Hierarchies and Interfaces

12. Arrays, `ArrayLists`, and Multidimensional Arrays

13. `MineSweeper` Programming Project

14. Searching and Sorting

15. Ethics in Computing

16. `GridWorld` Case Study

17. AP Exam Preparation

18. Final Projects

4 Resources

4.1 Textbook

The textbook for this course is *Java Methods A & AB: Object-Oriented Programming and Data Structures* by Maria Litvin and Gary Litvin. The course problem sets will have required reading from the textbook and **you are responsible for doing the reading!** In fact, if you do not read the required sections in the textbook before attempting to complete parts of each problem set, you will find yourself spending far too much time figuring out how to answer the questions. The textbook is clearly written and succinct, so you won't have to read many pages to learn what is necessary.

4.2 Oracle's Java 7 API — and the AP subset

The entirety of Java's built-in libraries is explained in detail on Oracle's website[2]. The portion of Java that you'll necessarily need to succeed in this course and on the AP exam can be found here: http://www.cs.duke.edu/csed/ap/subset/doc/[3]. That link will prove very useful, so bookmark it!

4.3 Software

All of the software that will be used in this course is available for free, and much of it can be run on multiple platforms (e.g., Windows 2000/XP/Vista, Mac OS X, and Linux). Consult the course website at http://feromax.com for download details.

- **Eclipse** — a professional grade IDE[4]

- **Notepad++** — a programmer's text editor, useful for various C-style languages, including Java, PHP, and PERL

- **Oracle's Java Developer Kit (JDK)** — command-line tools that allow for compiling and executing Java programs; combined with a text editor like Notepad++, this makes for a very light and fast Java development platform

- **Processing** — an extension of Java that allows for rapid development of graphical programs

---

[2]see http://docs.oracle.com/javase/7/docs/api/

[3]Thanks to Owen L. Astrachan for his work to whittle down the massive set of Java libraries' APIs into a far more manageable set for AP students!

[4]Integrated Development Environment

4.4 Computers

4.4.1 School Computers

Our class will generally be meeting in the AOIT computer lab in Room 127. During lunch periods and after school, computers in my classroom and/or Room 127 will generally be available for your use.

4.4.2 Home Computers

If you have a computer at home, you are strongly encouraged to set up the course software on it. Setup instructions for Eclipse and the Oracle JDK on Windows will be provided later. Generally speaking, a PC with an Intel Pentium 4 processor (or equivalent) and at least 1GB of RAM will be sufficient to run Eclipse (though we have run it on less at school).

4.4.3 Managing Your Data Between Computers

You are responsible for managing your files! While at school, you will be able to refer to your saved files on the APCS server, `apcs02`. But when you go home and continue working, you will no longer have access to the server. You will need a system for making your files accessible all the time. It's strongly recommended you adopt at least one of these strategies:

- Keep a copy of your working files on a flash drive. As you make changes at home, make sure your flash drive is up to date and carry the drive back to school.

- Email your working files to yourself before you leave school and again when you finish working at home.

- Upload working files to a cloud storage service like Google Drive, Dropbox, or Copy.com and make sure your home computer synchronizes your changes to the cloud service so you may retrieve your homework while at school.[5]

4.5 Course Website and Email List

A website has been developed for this course and is available at `http://feromax.com`. There you will find all handouts, problem sets (plus errata[6]), and course news. The site will also allow you to maintain contact with me and your classmates, both during and after school hours. You can ask questions in a forum manner and help others with their questions.

Once students have registered for accounts on the website, a course mailing list will be created. That email list will be used to broadcast changes to due dates, reminders, and notifications of when grade updates are posted on School Loop.

---

[5]SFUSD's Internet filtering may block access to one or more of the mentioned cloud storage services and that is subject to change over time.

[6]An erratum is a correction, usually to a book. Errata is a set of corrections, and we use the term in computing to describe corrections to errors in manuals or security updates for software.

5 Teacher's Expectations

It is expected that you will be learning from various places, including lessons, reading, and your own tinkering with code. The degree to which you rely on each is a matter of your own learning style. I respect that everyone has their own style, but I also expect that you draw experience from *all* of those sources.

When in class, it is extremely important to be paying attention during lessons and explanations. That means that you might have to stop work on a problem momentarily to hear what I say. *And it certainly presupposes that you are not using a computer for any kind of non-class activities.* Your attention is considered a matter of educational necessity and politeness to me and your peers; your lack of it will be considered a failure to gain participation credit when it comes time to compute your grade.

All students agree to. . .

- respect your classmates, teacher, and equipment;

- be on time and working immediately, which entails connecting your workstation to your work folder on apcs02, starting Eclipse, loading the day's lesson slides, and starting the *Do Now*;

- make progress on the current problem set on a **daily basis**, even in cases when there is no strict due date for interim sections;

- read all assigned sections from the textbook, which are listed throughout the problem sets;

- supply your own effort for your own work (i.e., not giving others your work or vice versa);

- provide *appropriate* assistance to your peers (refer to prior bullet point!), as not everyone will learn all topics at the same speed or understand abstract concepts in quite the same way;

- operate computers in an ethical manner, especially when personal information is involved; and

- LEARN!

6 Grading

Grades will typically be updated on a one- to two-week basis, with grades being posted on School Loop.

6.1 Points Allocation

Problem sets will typically have point values in the $100-200$ range, depending on the number of days each takes to complete. Quizzes will typically be worth $50-100$ points, and will be administered *at least* once during each six-week grading period.

6.2 Overall Grade Computation

All assignments, projects, and quizzes will be given a point value. The formula for computing your overall grade is

$$\text{Overall Grade } (\%) = 90 \left(\frac{\text{total points earned}}{\text{total points possible}} \right) + 0.1 \,(\text{ effort grade as percentage })$$

Thus, 90% of your grade comes from point-valued work, while the remaining 10% is based on effort — attendance, behavior, participation, and citizenship.

6.3 Late Work

Students who do not meet due dates for problem sets or project assignments receive automatic grade deductions, the values of which are determined by two factors:

- number of days late

- total number of points the assignment is worth

A typical point deduction is on the order of $15-20\%$ per day.

7 Problem Sets

7.1 Overview

Just as formal assessments measure what you have learned, formative assessment — in the form of *problem sets* — will let me measure how your learning is progressing and how much effort you are putting forth all along.

At any given time, there will generally be a problem set you are expected to be working on. Generally, lessons will conclude with a listing of problem set sections you are expected to work on and complete for homework. Expect that some class time will be used to check up on your problem set progress; this is how you prove that you are doing your homework all along before the final due date for a given problem set. *Checkpoint scores* may be given for certain problem set sections, and these values will be used when tallying up your final problem set score.

7.2 Structure

Problem sets are generally structured in this manner:

- *Introductory text* is provided to prepare you for the material that will be presented. Sometimes, this text is above and beyond what your textbook provides, so it is important that you take the time to read all I have written.

- *Required reading* tells you which sections of the Litvin textbook you need to read **before** attempting to proceed further in the problem set.

- *Book problems* are those problems from the Litvin textbook that you are required to work on.

- *Teacher-written problems* are those problems that either don't appear in the Litvin textbook or are written in a different manner (that is hopefully easier to understand).

- *Programming assignments* generally describe some programming task for which you will be asked to provide either (a) written source code, (b) demonstrate a working program, or (c) both. **These assignments** typically have a set point value and **are significant to your problem set grade!**

7.3 Quality of Work

Just as grammar, spelling, neatness, and overall presentation of a deliverable matter in industry (and academia!), those elements do count for your problem set grade. Sloppy work that isn't well written does not reflect well on your effort level. Just as this very document is well formatted and clearly written, I expect the same from you. Those who would judge you based on your work product — teachers, managers, prospective clients, etc. — are often gatekeepers to your future success. That said, it doesn't hurt to impress. (By the same token, one can go overboard by spending too much time on aesthetics and too little on content.)

8 Getting Help

If you are having difficulty with a particular concept or part of a problem set, you are highly encouraged to do the following, in order.

- Re-read the question in case you missed important details.

- Use debugging techniques you are taught in class: insert print statements to show variable state at various places, test tricky lines of code in a small program where you control the conditions, etc.

- Make sure you try to understand any compile-time errors that are generated.

- Look carefully, line by line, for tricky syntax errors: missing semicolons, unmatched curly braces, missing a capitalization, etc.

- If in the computer lab, ask a classmate to have a look at your code. When outside of class, use the web site's problem set posting as a place to post a question.

- Ask me.

Office hours (after-school and lunchtime tutoring) will be announced, and it's expected that you will come to them to get help with topics/questions that are confusing for you. This time is also good for getting programming assignments checked off in your problem sets.

9 Collaborative Work Policy

Part of becoming a computer scientist, software architect, developer, QA person, etc., entails working with peers in some capacity. I *encourage* students to work together to simulate that real-world experience. In addition to the preparation for what comes later, you will learn how to

interpret the code of others, what good comments in code are (and are not!), and the importance of good programming style. The most direct advantage is this: Working with others can streamline your time spent being stuck and debugging your code.

The main disadvantage of working with others can be the temptation to breach a code of ethics that I make my students uphold. For example, let's say you're having a hard time figuring out how to write a program that finds the nth Fibonacci number in a recursive manner. You ask your friend, who is also in the class, "Hey, what's wrong with my code?" After she has a look, she decides that it's all wrong and not really worth trying to fix. She suggests, "You might want to start over. Here, have a look at what I have."[7] Sounds innocent enough, but you can probably make a few small mental leaps between where this story ends and having lines of her code embedded within your own.

In these days of instant messaging, email, web forums, etc., opportunities to *borrow* code abound. Do yourself a favor and **make sure your discussions with others stick to methodology**, or **how to do something**, not adapting specific chunks of others' code.

10   Academic Honesty Policy

Universities have established policies for dealing with academic dishonesty, and you will be held to a similar standard while in this class. **Under no circumstances are you permitted to present anyone else's work as your own.** I understand that it can be frustrating when you can't get the right answer, but there is more than enough support available to help any student who is having difficulty. Penalties for academic dishonesty may include — but are not limited to — grade reductions, referrals to school administration, and notification being sent to the colleges to which you apply.

One contributing factor to academic dishonesty is falling behind on your work, a.k.a. PROCRASTINATION. If you fall too far behind, it becomes *impossible* to catch up without cheating. I would rather you come talk to me about the situation instead. First, I will respect your honesty. Second, I may be willing to make an accommodation to help you get back on track, like pushing back a due date for some agreed-upon grade penalty. Of course, if you have some legitimate issues preventing you from getting your work done, I will be willing to work out an arrangement with you provided that you do the responsible thing — find an adult you trust to help you, be it me, another teacher, a school counselor, or a Clinic counselor.

---

[7]Point for discussion: Is the student with working code, who is offering her code to a friend, guilty of *academic dishonesty*?

Parent Letter for AP Computer Science, 2013−2014

Dear Parent/Guardian:

I hope that you have had an opportunity to read over this syllabus and discuss it with your son or daughter. If you have any questions or concerns, please don't hesitate to contact me at **415-469-4090 x3319** or via email at <**mferraro@balstaff.org**>.

Grades for this class will be made available on School Loop (see <`http://balstaff.org`> for a link) and you will be notified of grade updates via email. Course materials will be posted to School Loop and/or the course website at <`http://feromax.com`>.

So that I may be in touch with you with grade updates and such, please provide your contact information to me by filling out the form below, tearing it off, and having your son or daughter return it to me. Thank you in advance!

Sincerely yours,
Michael Ferraro
Balboa High School

·· *tear here* ··

Parent Information for AP Computer Science, 2012−2013

Student's Name: _____ Period: _____

Your Name(s): _____

Home Phone: _____ Best time to call: _____

Work Phone: _____ Best time to call: _____

Cell Phone: _____ Best time to call: _____

May I phone you in English? [YES / NO]   If NO, which language? _____

Email Address: ⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚⬚

How often do you check email? _____

Signature: _____ Date: _____

Please write any comments/concerns you have on the back of this form. Thank you!