SPECIFICATIONS FOR FINAL PROJECT IN APCS

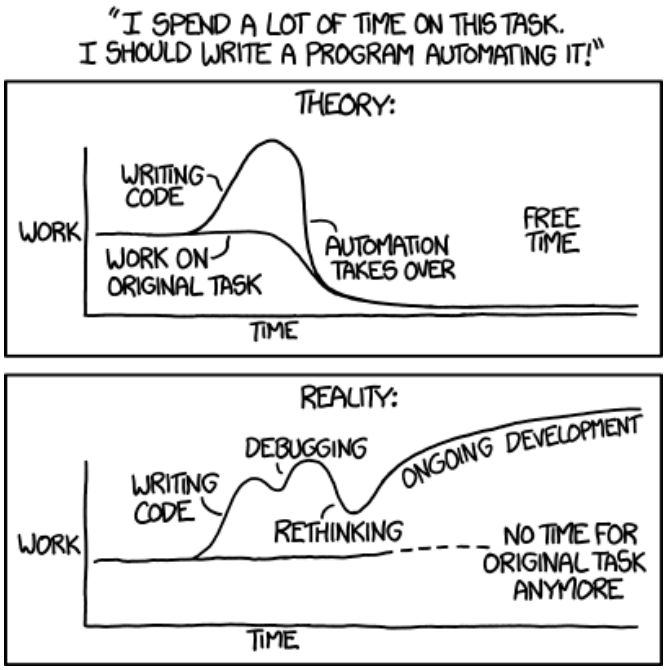
Due: Thursday June 06 & Friday June 07

So it's the end of the year. You've nearly taken your AP exam and you're wondering. . .what's next??? It's time for a final lab project!!!!

You may work in teams of up to four programmers. This project will take the place of a final exam and is worth 30% of your overall grade!

Your program may be constructed to do anything that you wish. I'll give some advice though. Don't bite off more than you can chew! You'll only have a few weeks to get your idea rolling and then pull the code together with your team. If you have a grandiose idea, try to construct it in a modular fashion. Start with a working "bare bones" skeleton of what you're planning to do and then add functionality to the working parts. This way, on deadline day you have something that is WORKING!!! If it doesn't do all the things you had planned, that's OK! This modular approach may help you to keep classes in mind while you build your program. Remember, you can flesh out your code with bells and whistles once you have a working architecture.

I'm looking for your final project to be a fairly involved piece of code. You may do extra research into graphics, networking, advanced data structures, artificial intelligence, simulations, operating systems, hardware drivers, and the list could go on and on. YOU'LL have to do the research to find ways to accomplish the goals you have. I'm not an encyclopedia in all aspects of coding. I can however, provide assistance on many topics and at least steer you to a good resource or in the right direction.



I would expect your program to work with many classes that you create from scratch. Look at the class interaction in the Elevens project. It's a decent example of how many objects can work together to obtain a particular goal. Have different group members work on different classes for your project. This will divide up the research work that you'll undoubtedly need to do.

A word of warning: If you're team is making a game, I'd suggest a strategy game of some sort. Focus more on game play using simple rules and allocate fewer of your resources to graphics. There is a pathway available for working with some of our amazing graphic artists here on campus!

Points:

100 WHAT YOU HAVE ON DEADLINE DAY IS WORKING

100 CODE – your code is broken into classes that make sense for what you're trying to accomplish. The classes interact well with one another. Code is traceable and it is evident who created what parts of the code. Be able to verbally provide information about why certain methods/ algorithms were chosen.

100 DOCUMENTATION – your code is well commented for aid in readability. In your final project workspace folder include javadocs that contain full documentation of your classes, interfaces, and how the program works. Also include ALL outside references that you used in your research.

100 ORIGINALITY – This is a project which should reflect the level of sophistication of students of your caliber. This category will reflect your design ideas, your approach for solving the problem you choose, and any intangibles.

See the following rubric

	APCS Project Rubric			
Objectives	100	80	60	0
Does it run???	• Code runs with no major crash incidents. There may be minor "glitches" but overall run is smooth.	• Code runs with numerous minor "glitches". May have a major crash when "pushed".	• Code must be "nursed" through trial to avoid crashing.	• "You mean it was supposed to run today??"
Code	• Effective Decomposition. •Good "Driver" class and class interaction. •Traceable and author credit given.	• Semi-effective Decomposition. •Good "Driver" class and some class interaction.	•Little Decomposition. •Has "Driver" class with little class interaction.	• "I thought we could do everything in main() !!!"
Docs	•Code has helpful "inline" comments to aid readability. •Project includes full javadoc build. •Project includes research references. •All final documents are burned onto CD or USB stick.	•Code has helpful "inline" comments to aid readability. •Project includes full javadoc build. •Project includes research references.	•Code has helpful "inline" comments to aid readability. •Project includes full javadoc build.	• "Nooooo!! We had to comment this thing??!!"
Originality	•OK, so how do we market and sell this thing? •Good design choices and interface. •Effective, fun, fast?????	•Impressive. Some more work, and we could make some money! •Acceptable design and interface. •Effective, fun, fast???????	•Where's the bells and whistles that will sell this thing? •"Design is for fashion shows." •Effective, fun, fast???????	•"This is OK, but my pet gorilla could code this!"