**AP Computer Science Project**

This project will require you to design and implement a Java-based game, simulation or significant Java project. It will be completed in 3 phases. The first phase will consists of a project proposal that you will present for review. After a debriefing meeting, a second draft will be submitted that includes any changes/revisions to your project design. Phase two will require you to write a working Java program adhering to code requirements listed below. The final phase will require you to present your project to the class. Each of the phases is described in detail below.

**Stage I - Proposal**

A THREE-page proposal will be submitted. The first page will be a full page single-spaced TYPED description of your project describing the basic operation of the game/simulation/project. The second page will be a hand-drawn or computer generated picture/screenshot that illustrates what a typical screen(s) might look like in your program. The third page will be a UML class diagram describing the classes in your project (public/private instance variables and methods). At the debriefing meeting, we will discuss your design and possible changes to the class design and/or implementation. This does not have to be a final design; you may (and probably will) make design changes. The SECOND proposal should include any changes/additions/deletions that you make to your first proposal.

**Stage II – Java Program**

The program must include the following 12 items.

1. ~~At least two relational/comparison operators (==, !=, >, >=, <, <=) and two logical operators ( &&, ||, !)~~
2. ~~At least two “if-then-else” statements.~~
3. At least one of **EACH** of the following: “~~for-each loop~~”, “~~while loop~~”, and “~~for~~” loops.
4. At ~~least ONE~~ **~~student-designed~~** ~~interface~~ and ~~THREE~~ **~~student-designed~~** ~~classes (one of which MUST be abstract), not including your driver class (e.g. “OthelloDriver.java”) – which brings the total to FIVE.~~
5. ~~Interaction between all four student-designed classes/interfaces in your project must be implemented. Specifically, each class must call methods from or be composed of AT LEAST one other student-designed class/interface in your project.~~
6. An Inheritance hierarchy must be implemented with the student-designed classes.
7. Polymorphism must be implemented with the **student-designed** classes.
8. ~~Class ArrayList must be used in at least ONE~~ **~~student-designed~~** ~~class and it MUST be traversed through AND accessed via a for loop OR a for-each loop.~~
9. Comments explaining logic and operation of program at “key points” (e.g. special algorithm to determine possible next moves in Chess, to check winners in Connect Four, to follow Pac-Man around the screen, etc.)
10. ~~Meaningful variable/class names throughout (class, methods, variable name, instance variables, etc.)~~
11. Including “javadocs” for each of your methods in your project (each method should be commented according to javadoc specifications)

Some ideas for projects are:

- Bowling - PacMan

- Hangman - Yahtzee

- Monopoly - Backgammon

- Jeopardy - Othello

- Wheel of Fortune - (Chinese) Checkers/Chess

- Basketball games - Natural Selection Biology

- Quizlet - Google Feud (Liberty Feud?)

- Risk - Life

- Traffic light simulation - Airline reservation system

[**http://thecubscientist.com/APCS/indexAPCS\_Projects.html**](http://thecubscientist.com/APCS/indexAPCS_Projects.html)

**Stage III – Presentation**

A Powerpoint/Keynote/Google Slide presentation will be given in class. The presentation shall have, at minimum, the following slides:

* Title
* Description of program operation (or how game is played)
* Demonstration of Program (Task switch from your slide show presentation to your program using Alt-Tab or Cmd-Tab)
* UML Diagrams for each class
* Use of classes/objects in project - elaborate on how classes represent **physical objects** in your program **(be prepared to justify class names, class data member names, class method names…)**
* Description of class interaction **(be prepared to discuss how each class interacts with the other classes)**
* Description of use of an inheritance hierarchy **(be prepared to justify structure)**
* Description of use of an interface **(be prepared to justify its use with other classes)**
* Description of use of polymorphism **(include a code snippet that demonstrates polymorphism)**
* Special features implemented - elaborate on tricks/special things you did
* Known bugs in program
* Citation of “second-party” code used in program (be able to explain code)
* Conclusion - Summary of what you thought of writing the program
  + Difficulty level,
  + "Fun" level,
  + Your evaluation of the final product,
  + What you learned (be specific)
* Questions? (this is simply a slide that says “Questions?” that keys the audience for any questions they might have)

The items that should be turned in with your final presentation are as follows:

- Source Code (your name should appear in the first line of each source code file)

- Class Diagrams

- Keynote/Powerpoint/Google Slide presentation slides

[**http://thecubscientist.com/APCS/indexAPCS.html**](http://thecubscientist.com/APCS/indexAPCS.html)