APCS FINAL PROJECT

Your final project for APCS will pull together many of the design concepts and programming skills that you have learned this year. This project is worth 10% of your entire semester grade. You will put together formal design specifications, divide up the program between you and your partners, do the software development, test each portion of the project, and present your final program to the class during the last two weeks of school. (SENIORS should be prepared to present their projects first, so that final grades can be submitted.)

In groups of **2 or 3** students (from the same period) you should:

1. Select a project and write up a project plan. The plan must be created using the provided templates.
2. Provide weekly status reports utilizing the provided templates.
3. Develop each section of the project separately. Demonstrate your intermediate functional test results.
4. Integrate (put together) all the program parts. Test, debug and revise as necessary.
5. Present your final product to the class in a 10-minute final presentation.

You may choose your own project, subject to approval. You will get more credit for completing all the milestones for a simpler project than **not** completing a complicated one. All projects must be written in Java.

Some project possibilities include:

POSSIBLE PROJECTS:

- Game (person vs. computer AI) (i.e. Poker, checkers, Battleship, Yahtzee, Connect 4)
- Simulation (i.e. enhanced GridWorld Case Study)
- Smart phone/tablet app (must be written in Java)
- Numerical analysis
- Large Scale Number Crunching (i.e. modeling)
- Business Application Program (i.e. SafeTrade, manufacturing tracking, inventory control)
- Home Office Application
- Other ideas…see me for approval

PROJECT GRADE: This project is worth 10% of your entire semester grade. The project points are broken down in the following way:

40% Project Documentation (using provided templates)

20% Final Class Presentation

40% Program functions as specified (points vary with each group's specifications and includes points for the required data structure). Includes Functional Test Programs, results and demonstration

EXTRA CREDIT: Up to 5% extra credit is available for groups that choose to investigate and build a graphical user interface and/or implement networking for your program.

PROJECT DOCUMENTATION (40 points): Your project should provide all of the deliverables specified in Task Summary and Timeline. The project documentation should include:

(1) a copy of all deliverable documentation

(2) a listing of your final program, test programs and testing results

(3) detailed design specifications for all classes generated from the Javadoc comments in the source files

(4) user manual and post project reflections.

(5) An electronic copy of all materials (reports, code, presentation, etc.)

NOTE: This materials must be complete (5 points – all parts there), accurate, organized, and easy to read (5 points). This means that your programs must be logically organized, well structured, and appropriately commented for full credit.

FINAL PRESENTATION (20 points): Your final presentation should

(1) Present the final project specifications (data structures, performance requirements and algorithms) to the class (5 points).

(2) Demonstrate the use of your program (5 points).

(3) Discuss the development of your software, its testing and the specification changes you made during this process (5 points).

NOTE: You may be as straight up or as creative as you like with your presentation as long as it includes the above items. (PowerPoint, skits, videos, songs,…)

IT WORKS (40 points): The program functions as specified and implements Object Oriented Design methodologies and principals. Points will vary with each group's specifications and includes points for the required data structure as follows:

Your group must choose at least **one** of the following data structures to use for your project:

- Multi-dimensional arrays w/ Recursion
- Linked List (Double, Circular, etc.)
- Binary Tree
- Stack
- Queue
- Priority Queue
- Sets/Maps
- Hashing
- Heap

Testing (15 of 40 total points): After you decide what functionality should be developed, you must decide how to test each method to make ensure that it works as required and specified. This testing should **include JUnit test programs** that call your methods with specified parameters (test data) that test all its functionality. Don't just test the usual cases – remember to investigate the "boundary conditions." You should decide what these tests will be even before you develop the methods themselves and include them in your project proposal. If you discover that you need to modify the functional specifications, you will need to discuss the changes with your group and rewrite your overall specifications. **Keep a log of the specification changes you make to include in your final written report and presentation. Make sure these changes are approved or you may loose project points.**