Summative Project Expectations

- **Topics/Project Goals:** For the most part, this is pretty much open to you. At a minimum, you should consider:
    - **A project should satisfy the following requirements:**
        - It should be interesting.
        - It has a clear goal.
        - It has clear results.
        - It demonstrates the breadth and depth of your programming knowledge.
        - It is not too ambitious.
        - It can be carried out in a 2 week period.
    - **Going *beyond* the topics as discussed in class. Some examples of this could be:**
        - Create a program that makes life easier or solves a problem(real life or academic).
        - Finish out an implementation of a fully coded game. (Can use RPG from CS1 as a springboard)
        - Exploring other languages/platforms that use OOP languages, such as building an app for Android execution or an application for Raspberry PI, or building an app/PC game in Python, Ruby or C#.
    - **The project should not only explore a new application of the topics we've studied, but should be a *teachable* project:**
        - You will present your project to the class, showing the steps/stages you went through to complete the project. Expect your total presentation, including demonstration, activities, and Q&A to take 20-25 minutes.
        - You should be able to compare/contrast/exemplify the methods used in creating your project with the topics we have covered so far this semester (logic, OOP concepts, rules and structures.)
        - Your code should follow best practices.
        - If your project can be executed/used/played with by classmates, have a way for them to be able to do so.
        - You should provide some resources / instructions for your classmates that help them understand how to create a project similar to this on their own.
        - You should provide at least one activity for your classmates to do that will show them how you completed the project, or how something works as part of your project.
        - Be prepared to answer questions about your project from both your teacher and your classmates.
        - Provide proper documentation your team will need the documentation. Multiple students work on the same program and frequently you will look and help to debug program segment created by other team members.

- **Expected Deliverables**
  - Short(1-2 page) project plan
    - Name: a descriptive name
    - Synopsis: One sentence summary
    - Objective: ½ page - 1 page on *what* you want to achieve
    - Approach: ½ page - 1 page on *how* you want to achieve it
    - Expertise: What do you need to know to do this?
    - Risk: What could go wrong.
    - Results: what are the results you expect?
  - UML Diagram
  - Source code
  - Self-Executable .jar
  - Presentation
    - Focus on goals and benchmarks (achievements).
    - Not too many details.
    - Highlight what was interesting or unexpected.
    - What did you learn?
    - Demonstration
    - Q & A

Timeline:

- May 8-9: research possible projects or find tutorials to help you with your coding implementation online. Set up work environment/project management.
- Be prepared to make a decision on your project concept by Thursday, May 10.
- Project implementation plans is due Friday, May 11. (daily grade)
- UML Diagram is due Monday, May 14. (daily grade)
- You will have 11 days to prepare your project and presentation. Presentations will begin Friday, June 1st. (peer evaluations will be a daily grade, teacher evaluation will be a test grade)