

AP Computer Science Project I

Mr. Lew

Fall Semester

This project will require you to design and implement a Java-based game or simulation. It will be completed in 3 phases. The first phase will consist 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 in a full period presentation. Each of the phases is described in detail below.

Stage I - Proposal

A THREE-page single-spaced proposal will be submitted on the due date noted on the planner. The first page will be a TYPED description of your project describing the basic operation of the game/simulation. 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 computer generated 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 13 items.

1. At least two relational (==, !=, >, >=, <, <=) 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. Composing one class of another is class composition (recall the Time, Flight, Trip AP Problem).
6. An Inheritance hierarchy must be implemented with the student-designed classes (i.e. not the ActionListener interface). Note: if using GridWorld, you may NOT alter any of the "Actor" classes in the standard package (e.g. Actor, Bug, Critter, etc.), although you may inherit from these classes and override methods in these classes. You may, however, modify the Grid class in GridWorld package if you wish to modify the operation of the GUI.
7. At least one student-designed interface must be implemented in the project.
8. Polymorphism must be implemented with the **student-designed** classes.
9. 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.
10. 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.)
11. Meaningful and variable/class names throughout (class, methods, variable name, instance variables, etc.)
12. Including "javadocs" for each of your methods in your project (n.b. each method should be commented according to javadoc specifications)

13. Use of JOptionPane for user input OR the use the GridWorld GUI interface.

Some ideas for projects are:

- Blackjack (or any card game)
- Hangman
- Monopoly
- Jeopardy
- Wheel of Fortune
- Basketball games
- Football games
- Risk
- PacMan
- Bowling
- Yahtzee
- Backgammon
- Othello
- (Chinese) Checkers/Chess
- Connect Four
- Missile Command
- Life

The Final Project is due on your assigned lottery date. **THERE WILL BE A 15% deduction of points for each day that the project is late. There will be a 5 point deduction for each typographical or grammatical error.**

Stage III - Presentation

A Powerpoint/Keynote presentation should be given on your lottery date. The presentation shall have, at minimum, the following slides:

- a. Title
- b. Description of program operation (or how game is played)
- c. Demonstration of Program (Task switch from your slide show presentation to your program using Alt-Tab or Cmd-Tab)
- d. UML Diagrams for each class
- e. 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...**)
- f. Description of class interaction (**be prepared to discuss how each class interacts with the other classes**)
- g. Description of use of an inheritance hierarchy (**be prepared to justify structure**)
- h. Description of use of an interface (**be prepared to justify its use with other classes**)
- i. Description of use of polymorphism (**include a code snippet that demonstrates polymorphism**)
- j. Special features implemented in program - elaborate on tricks/special things you did
- k. Known bugs in program
- l. Citation of "second-party" code used in program (**be able to explain code; not including GridWorld GUI**)
- m. Conclusion - Summary of what you thought of writing the program
 - i. Difficulty level,
 - ii. "Fun" level,
 - iii. Your evaluation of the final product,
 - iv. What you learned (be specific)
- n. Questions? (this is simply a slide that says "Questions?" that keys the audience for any questions they might have)

Source code and Powerpoint presentation should be submitted in a 1" THREE RING BINDER at the beginning of class before you begin your presentation. The items that should be turned in are as follows:

Printouts

- Source Code (your name should appear in the first line of each source code file)
- Javadocs (printed in landscape mode)
- Computer-generated UML Diagrams
- Keynote/Powerpoint presentation slides (2 slides per page MAX)

In your source code printout, draw a box around one instance of each code snippet that satisfies one of the 13 code requirements. Therefore, you should have 13 sections of boxed code that represent the 13 **required** sections of code (i.e. one box around an if statement, one box around the use of an ArrayList, etc). A hardcopy of the javadocs generated for your project should be placed after the source code printout. BlueJ will generate these automatically.

Email Zip file to Mr. Lew at mlew@loyolahs.edu. Zip file will include all of the item listed in the "Printouts" section above.

Pointers for your final project:

1. **START EARLY!! Don't wait until after Christmas to begin your project!!**
2. **Make sure you have all printouts on the presentation day!!**
3. **Test your program on the presentation computer before the presentation date.**
4. **Using System.out.println() statements to display values during the debugging process.**
5. **MAKE BACKUPS OF YOUR WORK!!**
6. **And, of course,...**

HAVE FUN :-) !!