

AP Computer Science Project I

Mr. Lew

Fall Semester

1. This project is worth 570 points
2. The program must include the following 15 items. Each item is worth 25 points.
 1. Wrapper classes – Integer AND Double (with intValue(), doubleValue() methods)
 2. Relational (==, !=, >, >=, <, <=) and Logical Operators (&&, | |, !) - at least 5 total
 3. One or more “if-then-else” statements
 4. One or more “for-each” AND “while” OR “for” loops
 5. At least ONE **student-designed** interface and THREE **student-designed** classes (one of which MUST be abstract), not including class Test in “TestDriver.java” – which brings the total to FIVE.
 6. Interaction between all four classes/interfaces (i.e. each class must interact with AT LEAST one other class (not-including class Test in “Test.java”)
 7. An Inheritance hierarchy must be implemented with the **student-designed** classes (i.e. not the ActionListener interface)
 8. At least one interface must be implemented with **student-designed** classes.
 9. Polymorphism must be implemented with the **student-designed** classes.
 10. 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.
 11. 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, etc.)
 12. Animation OR graphical Interface (JOptionPane) must be implemented
 13. Meaningful variable names
 14. “javadocs” for each methods in your project (n.b. each method should be commented according to javadoc specifications)
 15. Program should be “jar’ed”

Subtotal = 350 points

With pencil, place a box around each section of code that satisfies a requirement. Therefore, you should have 13 sections of boxed code that represent the 13 **required** sections of code. A hardcopy of the javadocs generated for your project should be placed after the source code printout.

3. A TWO-page single-spaced TYPED description and computer generated UML class diagrams describing your project is due **AT THE BEGINNING OF CLASS on “Proposal I” Due Date as shown on planner**. It should outline what your project will do (whether it be a game or some other program). At this time I will let you know whether the project is too simple or too complex for a final project. This does not have to be a final design; you may (and probably will) make design changes. The proposal is worth 50 points.

Subtotal = 400 points

4. A SECOND TWO page single-spaced TYPED description and computer generated UML class diagrams describing your project is due **AT THE BEGINNING OF CLASS on “Proposal II” Due Date as shown on the planner**. This SECOND proposal should include any changes/additions/deletions that you make to your first proposal. The second proposal is worth 50 points.

Subtotal = 450 points

5. Some ideas for projects are:

<ul style="list-style-type: none"> - Blackjack (or any card game) - Hangman - Monopoly - Jeopardy - Wheel of Fortune - Basketball games - Football games - Risk 	<ul style="list-style-type: none"> - Bowling - Yahtzee - Backgammon - Othello - (Chinese) Checkers/Chess - Connect Four - Missile Command - Life
---	--

6. The Final Project is due on your assigned lottery date. **ONE HUNDRED (100) POINTS WILL BE DEDUCTED FOR LATE PROJECTS.**

7. A Powerpoint presentation should be given on your lottery date. The presentation will be worth 120 points. 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 Powerpoint presentation to your program using Alt-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 talk about which classes create each of 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**)
 - 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)

Subtotal = 570 points

8. Source code and Powerpoint presentation should be submitted in a 1" THREE RING BINDER (see example in class). The items that should be turned in are as follows:
 - A. Printouts
 - Source Code with complete comments and javadocs (printed in landscape mode)
 - javadocs
 - Computer-generated UML Diagrams
 - Keynote/Powerpoint presentation slides (print 6 slides per page)

- B. Email Zip file to Mr. Lew at mlew@loyolahs.edu. Zip file must include the following:
- Source Code with complete comments
 - javadocs
 - Computer-generated UML Diagram
 - Keynote/Powerpoint presentation

Any deviation from these guidelines will result in a 100 point deduction in points.

Point Distribution

Project Proposal I	50 points
Project Proposal II	50 points
Requirements 1-15	350 points (zip file and printout is required for full credit)
Powerpoint Presentation	120 points (zip file and printout is required for full credit)

(There will be a -5 points EACH misspelling or grammatical error)

Total points 570 points

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 :-) !!