Project – Computer Game or Simulation

Your cumulative project is to design, write, and present a Java-based game or simulation individually or in groups of two.

Stage I – Planning
After thoughtfully selecting a game or simulation, prepare a written proposal of your project. Your written proposal should contain: (1) a typed (one-page single-spaced) description of the basic operation of your program, (2) hand-drawn or computer-generated picture(s)/screenshot(s) that shows what the screen might look like during the execution of your program, and (3) a UML class diagram illustrating the classes and inheritance structure of your program (see attached example).

Your proposal must be submitted, by the end of class, on Friday, January 11.

Stage II – Program Development
You are free to make your program as complex as you like. However, at a minimum, your program needs to contain the following items:

1. two relational (==, !=, >, >=, <, <=) operators
2. two logical (&&, ||, !) operators
3. two "if-then-else" statements
4. one "for" loop
5. one "while" loop
6. one "for-each" loop
7. a total of five student-designed classes
    a. one driver class
    b. four classes
8. interaction among the four non-driver student-designed classes: each class must extend or call methods from at least one of the other classes
9. implementation of an inheritance hierarchy with the student-designed classes (not *extends JPanel*)
10. the Class ArrayList must be used in at least one student-designed class and it must be traversed through and accessed with a "for" or "for-each" loop
11. comments explaining program logic and operation at important points of the program
12. meaningful variable, method, and class names (including camelNotation)
13. "javadocs" for each of your methods (see attached example)
14. use of user input

The following project ideas are included to encourage your creativity.

- Blackjack, Poker, or any card game
- Monopoly
- Football, Basketball, Baseball, etc.
- Backgammon
- Chess
- Othello
- Hangman
- Jeopardy
- PacMan
- Connect Four
- Risk
- Wheel of Fortune
- Yahtzee
- Checkers

Stage III – Presentation
Your presentation is to be given on your selected presentation day; it should last a maximum of 15 minutes (not including questions). **There will be a 20% reduction in your grade if you are not able to give your presentation on your assigned day.**

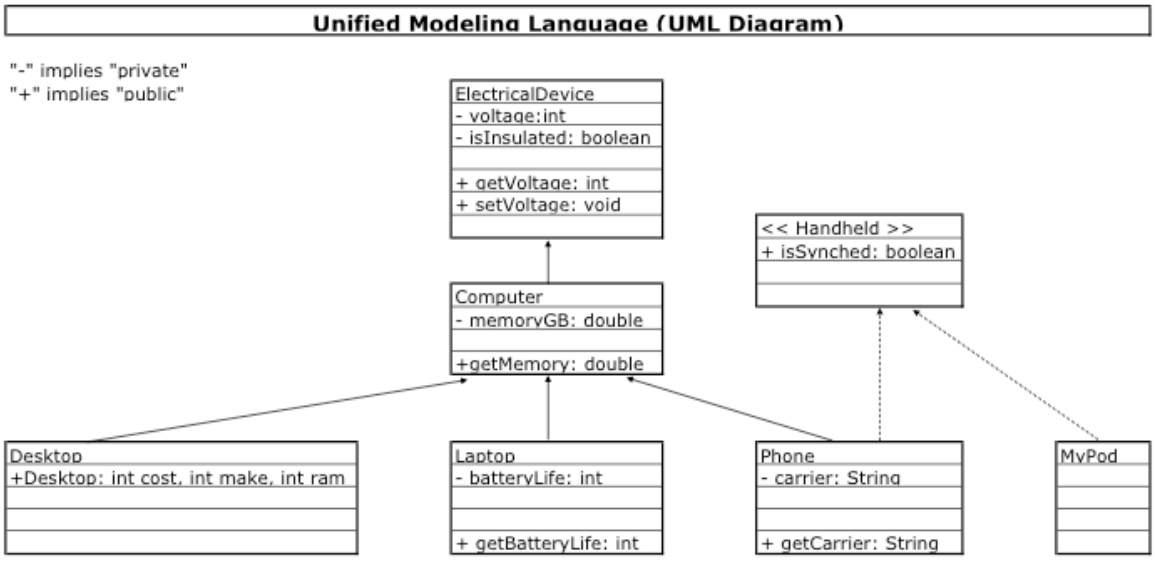
The presentation should be made using PowerPoint, Keynote, or other presentation software unless previous permission for an alterative method is granted. At a minimum, the presentation should have the following items:

a. Title
b. Description of program operation
c. Demonstration of program
d. UML Diagram
e. Use of classes/objects in your program
f. Description of class interaction
g. Description of your inheritance hierarchy
h. Special features/tricks you used
i. Known bugs in your program
j. Citation of "second-party" code used in your program (*be able to explain code*)
k. Conclusion – briefly summarize your experience of creating a program in regard to:
    a. Difficulty level
    b. Fun level
    c. Self-evaluation of your project
    d. Lessons/things you learned (be specific)
l. An opportunity for questions

Submission
On the day of your presentation, you will also submit electronic files of your project, including your code and your presentation slides, to holmteaching@gmail.com.

Project Tips
1. Don't wait! – your presentation day will arrive quickly
2. Plan ahead - use top-down development and pseudocode to plan your project
3. Keep your methods simple (each method should do one thing well)
4. Comment as you code to allow yourself and others to provide assistance more easily
5. Make use of System.out.println() statements to debug
6. Make frequent backups of your work
7. Ask questions if you get too stuck
8. Enjoy yourself!

UML Example


Unified Modeling Language (UML Diagram)
"-" implies "private"
"+" implies "public"
ElectricalDevice
- voltage:int
- isInsulated: boolean
+ getVoltage: int
+ setVoltage: void
<< Handheld >>
+ isSynched: boolean
Computer
- memoryGB: double
+getMemory: double
Desktop
+Desktop: int cost, int make, int ram
Laptop
- batteryLife: int
+ getBatteryLife: int
Phone
- carrier: String
+ getCarrier: String
MyPod

Javadocs Convention

```
/**
* Include a description of the method here
* description continued…
* description continued…
* @param describes the first parameter in the method
* @param describes the second parameter in the method
* @return describes what the method returns
* @throws describes any exceptions that are thrown
*/
```

Below is an example from the Student Class

```
/**
 * This method takes three grades and calculates the grade point average.
 * @param grade1 The first grade
 * @param grade2 The second grade
 * @param grade3 The third grade
 * @return The calculate grade point average
 */
public double calculateGPA( double grade1, double grade2, double grade3)
{
    myGPA = (grade1 + grade2 + grade3)/3.0;
    return myGPA;
}
```

The following screenshot is from the BlueJ Documentation for the Student Class

Method Detail

calculateGPA

```
public double calculateGPA(double grade1,
                           double grade2,
                           double grade3)
```

This method takes three grades and calculates the grade point average.

Parameters:

`grade1` - The first grade
`grade2` - The second grade
`grade3` - The third grade

Returns:

The calculate grade point average