Assignment #1 – Basic Lingo Game

Prepare to work on this assignment by creating a work folder called assign1 in your w-number folder. Call your program Lingo#.java, where the # is whichever version you are working on at the time. You must start with version 1 (Lingo1.java) and get it working before moving on to version 2 (Lingo2.java), etc.

Rules

Lingo™ is a popular game show on the Game Show Network, with simple rules. Players are given the first letter of a randomly chosen 5-letter word, and must guess the remaining letters by guessing words. For each guess, any letters that are in the word and are in the correct position are circled. Any other letters that are in the word, but are in an incorrect position are boxed. Then the blanks are rewritten, with any circled letters included in the correct spots, and the player guesses again. If the player figures out the word within 6 guesses, they win. If they fail (or anytime they guess an invalid word) they lose. For example:

Player 1 is presented with the first clue:

A __ __ __ __

Player 1 guesses “AGAIN”. The 2nd ‘A’ is in the correct position, so the blanks are filled to reveal:

A G A I N

And the clue is rewritten:

A __ __ __ __

Player 1 guesses “ALLOW”. The ‘W’ is in the hidden word, but not in that position, so the blanks are filled (covering the circled 2nd ‘A’) to reveal:

A L L O W

And the clue is rewritten:

A __ __ __ __

Player 1 guesses “AWAIT”. The ‘W’ is now in the correct position, so the blanks are filled to reveal:

A W A I T

And the clue is rewritten:

A W __ __ __

Player 1 guesses “AWARD”. This is the correct word, so the blanks are filled to reveal:

A W A R D

Player 1 wins!
**Version 1**

Write a program to play the basic Lingo game. Start by having a neutral player secretly enter a 5-letter word (just input the word, with instructions for the player to avert their eyes). Use dialog boxes for the input so that the input disappears after being entered. You may use a string for input of the initial word and the guesses, but you must process the guesses in a 5 character array. Then play the game as described above, but indicate that a letter is in the correct position by displaying an asterisk after it, and indicate that a letter is in the puzzle but in the wrong position by displaying a plus after it. The final box should either indicate a correct solution, or that the player is out of guesses (6 incorrect tries). So for the example above, the program would show the following sequence of dialog boxes:

Make good use of methods to modularize your design for this program.

*When you get this version of Lingo working, make sure you include the required comments (including descriptive comments for each variable and method in your program), and then commit it with the log message "Lingo version 1" (it should be named Lingo1.java). If everything is correct, you will earn 90 points (A level) for Assignment #1.*
**Version 2**

Modify version 1 to pick the starting word randomly from a large array of at least 50 strings which is initialized in the program source code, rather than inputing the word from a neutral player.

*When you get this version of Lingo working, make sure you include the required comments (including descriptive comments for each variable and method in your program), and then commit it with the log message "Lingo version 1" (it should be named Lingo1.java). If everything is correct, you will earn 100 points (A+ level) for Assignment #1.*

**Bonus Version**

Modify version 1 or 2 to use graphical output to show the sequence of guesses with correctly positioned letters shown in blue circles, and valid letters in incorrect positions shown in yellow squares. The entire sequence of guesses should be shown on a single screen. (Students unfamiliar with Java Task Force graphics are welcome to get assistance with making graphics when they are ready to try this version.)

*When you get this version of Lingo working, make sure you include the required comments (including descriptive comments for each variable and method in your program), and then commit it with the log message "Lingo version 1B" or "Lingo version 2B" (it should be named Lingo1B.java or Lingo2B.java). If everything is correct, you will earn 10 bonus points for Assignment #1.*

**What to turn in**

Commit versions 1 and 2 (with or without the bonus) in sequence as high as you can before the beginning of class on **Thursday, January 25, 2007**. Remember that you may commit multiple times for each version if you wish (and it is a good idea to do so for every major change or at the end of each work session). You don't have to make it all the way up to any version 1, but of course, if you do, your grade will be higher (as long as it is correct). Remember to have fun, and get help from the instructor when you need it!