# Assignment 3: Chapter 4, “Conditionals and Loops” (8%)

This programming project should be completed and submitted by the end of Week 8, and is worth 8% of your final grade. Please refer to the “Assignment Instructions” for details on the marking rubric and submission instructions.

1. Design and implement an application that plays the Hi-Lo guessing game with numbers. The program should pick a random number between 1 and 100 (inclusive), then repeatedly prompt the user to guess the number. On each guess, report to the user that he or she is correct or that the guess is high or low. Continue accepting guesses until the user guesses correctly or chooses to quit. Count the number of guesses and report that value when the user guesses correctly. At the end of each game (by quitting or a correct guess), prompt to determine whether the user wants to play again. Continue playing games until the user chooses to stop.

**Hint:** This will require a nest of loops—the outer loop to play each game until the user stops and the inner loop to prompt the user for each guess.

**Testing:** Include exhibits that demonstrate all the program’s features: multiple completed games, a game quit in progress, and how an input out of range is handled.

1. Design and implement an application that reads a string from the user, then determines and displays how many (of each) vowels (a, e, i, o, and u) appear in the entire string. Have a separate counter for each vowel. Also count and print the number of non-vowel characters. The logic must include a switch-case structure to determine which counter to increment.

**Hint:** Use the charAt method of the String class to examine each character of the input.

**Testing:** Build a test plan separately so you can predict and show that your program is working correctly. You need to determine a selection of inputs, such as something with no vowels, only a single letter “a,” etc. A key requirement is to prove that each counter is functioning independently.

|  |  |
| --- | --- |
| **Assignment Marking Criteria** | **Weighting** |
| **Correctness of solution:** Algorithm is implemented and produces correct results for the stated problem | /4 |
| **Testing:** Submission of test exhibits to indicate the solution works for a range of cases (e.g. minimum and maximum inputs) and handles unexpected exceptions | /2 |
| **Comments and documentation:** Source code contains comments that explain in plain English what the code is intended to do  **Note:** Javadoc style is **not** required. | /2 |