**Assignment 3**

**Due, Sunday, June 11, 2017 for maximum 100**

**Monday, June 12, 2017 for maximum 90**

**Tuesday, June 13, 2017 for maximum 80**

**Wednesday, June 14, 2017 for maximum 70**

**Assignment Scope**

1. Adding member variables or instance variable in a class
2. Generating getters/setters for member variables
3. Calling a method of a class using the reference object
4. Using Java API class **Random**
5. Use an enhanced **for** loop

**Deliverables**

To complete this assignment you must submit your **compressed Netbeans project** to Webcourses.

**Tasks and Rubric**

|  |  |
| --- | --- |
| Activity | |
| Boggle project |  |
| boggle package |  |
| Boggle.java | 1. Set the **ArrayList** member variable that stores the Boggle data equal to method call **shakeDice** in class **Board** |
| core package |  |
| Board class | 1. Add member variable of class **ArrayList** specifically using class **String** as the data type for the **ArrayList** to store the game data 2. Generate a getter for the game data member variable 3. Implement method **shakeDice** to do the following:    1. Loop through the 16 dice       1. Randomly select 1 of the 16 dice       2. Call method **rollDie** in class **Die**       3. Store the returned value in the **ArrayList** created above for the game data       4. Be sure to use each die only **once**, you will have to keep track of which die was used 4. Implement a method that will loop through the **ArrayList** of game data and display the board to the output window of the IDE (see figure 2) |
| Die class | 1. Implement method **rollDie** to do the following:    1. Randomly select one of the six letters of the die that will be used as the game data    2. Return that value |
| Boggle application |  |
| Test Case 1 | Test Case 1 passes |
| Test Case 2 | Test Case 2 passes |
|  | Source compiles with no errors |
|  | Source runs with no errors |
|  | Source includes comments |
| Total |  |

**Perform the following test cases**

|  |  |  |
| --- | --- | --- |
| Test Cases | | |
|  | **Action** | **Expected outcome** |
| Test Case 1 | **Project view** | Completed project view should look like figure 1 |
| Test case 2 | **Run application** | **shakeDice** output should be similar to figure 2 |

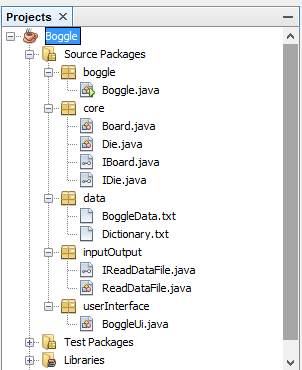
****

Figure 1 Project View

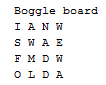


Figure 2 Output from method displayGameData()