Project 2

The goal of this project is to test your ability to use the knowledge of Java that you have obtained up to this point in the course.
The focus of this project is

- input validation,
- decision structures,
- looping structures,
- and random number generation.

Instructions

The goal of this project is to create a game in which the user enters whether or not a number posed to her is a prime number. The user will be playing against a computer opponent that will randomly guess yes or no. The program will keep score.

Program Functionality

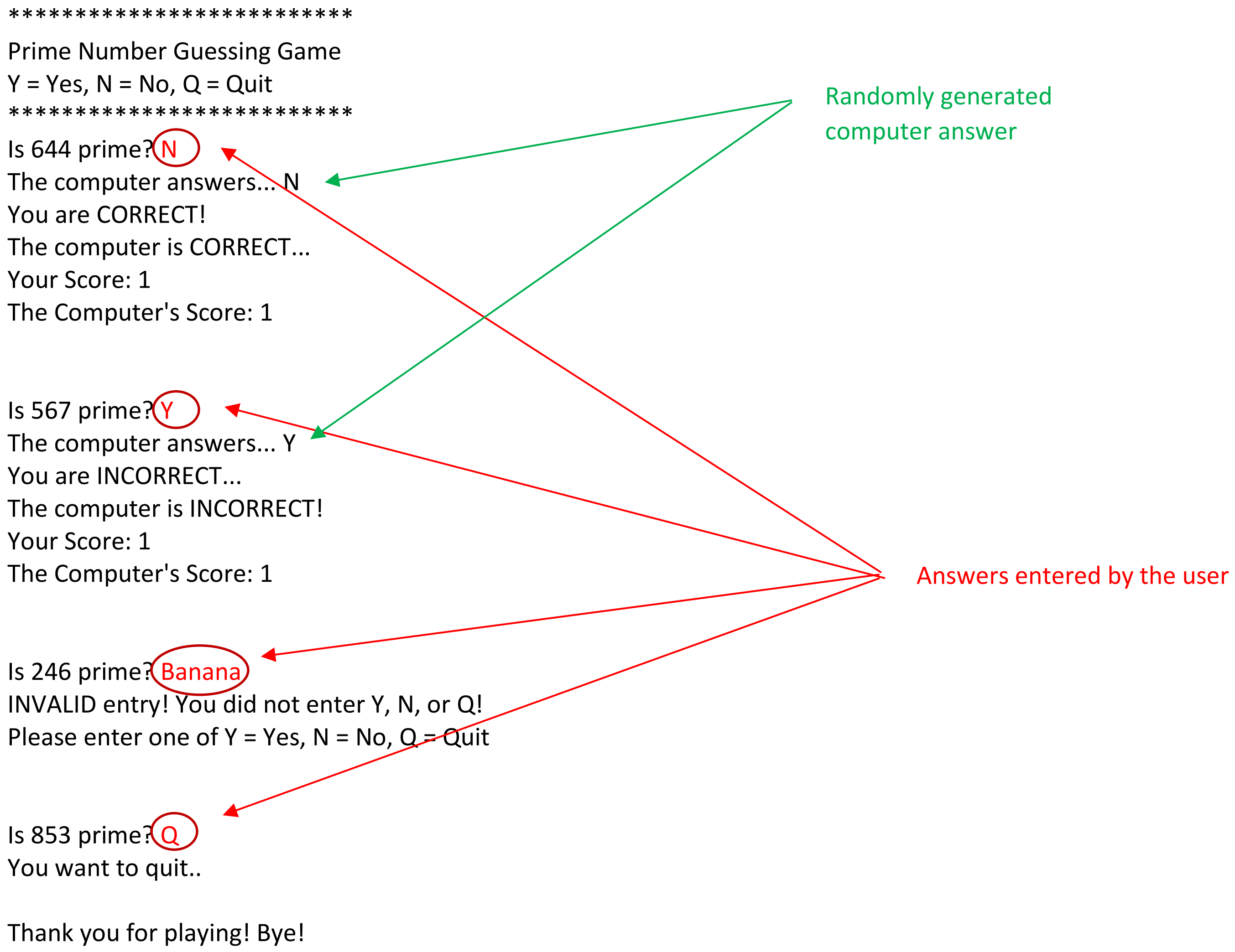
➢ Program Steps

1. Display the user a randomly generated integer (details in section below) and ask the user if it is prime.
2. Take in a “Yes”, “No”, or “Quit” response (details in section below).
   a. If the user enters a “Quit” response, the program should thank the user and the program should terminate.
   b. If the user enters anything besides a “Yes”, “No”, or “Quit” response, the program should display an error.
      i. And Start over from step 1.
   c. If the user enters a “Yes” or “No” response, this should be considered as their answer to the posed question.
      i. The computer opponent should choose “Yes” or “No” (details in section below) and its guess should be displayed.
      ii. The program should find if the randomly generated integer is prime.
      iii. Display if the user is correct or not.
      iv. Display if the computer is correct or not.
      v. Display the score of the user and the computer (details in section below)
      vi. Start over from step 1.

Details

- The randomly generated integer in Step 1 should be in the range 1-1000 inclusive.
- “Y”, “y”, “Yes”, or “yes” are all acceptable “Yes” responses.
- “N”, “n”, “No”, or “no” are all acceptable “No” responses.
- “Q”, “q”, “Quit”, or “quit” are all acceptable “Quit” responses.
- The computer opponent should randomly choose between yes and no. Each response should be as equally likely to be chosen. (i.e 50% chance of “Yes”, 50% chance of “No”)
- I would highly consider looping to find whether the generated number is prime.
- The score for both the user and the computer should start at 0, and 1 should be added to a player's score if she/it answers correctly. (So, if the user answers correctly on the first question and the computer does not, the user's score should be 1 and the computer's score should be 0).

Example Execution



```
*************************
Prime Number Guessing Game
Y = Yes, N = No, Q = Quit
*************************
Is 644 prime? N
The computer answers... N
You are CORRECT!
The computer is CORRECT...
Your Score: 1
The Computer's Score: 1


Is 567 prime? Y
The computer answers... Y
You are INCORRECT...
The computer is INCORRECT!
Your Score: 1
The Computer's Score: 1


Is 246 prime? Banana
INVALID entry! You did not enter Y, N, or Q!
Please enter one of Y = Yes, N = No, Q = Quit


Is 853 prime? Q
You want to quit..

Thank you for playing! Bye!
```

Randomly generated computer answer

Answers entered by the user

Project to do list:

□ **Step 1. Requirements**

- Make sure you completely understand the requirements and ask questions if you need clarification.
- Make sure that you pay close attention to detail and follow the instructions very carefully ☺

□ **Step 2. Design**

- Write a **pseudocode algorithm** or **a flowchart** for the solution to the problem (this should be completed before you start actual coding.)
- Remember: your algorithm is your road map – you will follow it when writing your code. DO NOT WRITE YOUR CODE FIRST AND THEN YOUR ALGORITHM/FLOWCHART! (It should NOT be the last thing that you do.)
- Test your algorithm before coding

□ **Step 3. Implementation**

- Create a **class called Project2**. Your project should be saved in a folder named **LastnameFirstnameProject2** (replace Lastname and Firstname with your own last and first name)
- Use your algorithm as a starting point for your comments throughout your program
- Write your program one step at a time, i.e. make sure one thing works before going on to something else.
- Complete the documentation of the application by inserting comments and adhering to programming standards
    - 1250 Coding Standards posted under Course Content in D2L
    - Pay particular attention to indenting, no word-wrapping when printed and comments

□ **Step 4. Testing**

- Test your program to make sure that it works!

□ **Step 5. Delivery**

- **Create a word document named Project2Design.docx** in the same folder as your project. At the very top of the document type YOUR name and CSCI-1250-xxx (where xxx is your section number) on the right-hand side.
- Insert your **Flowchart** or **algorithm** in the document (You can copy and paste your flowchart into the Word document)
- Place the completed Java Source Code (the .java file) in the same folder as a separate file in the same folder
- **Get three screenshots** of your code running and paste them in the Word document. **Use different input each time you run your program**.
- Zip up the folder containing your source code, flowchart/algorithm, screenshot(s). It should be named **LastnameFirstnameProject2.zip** (again, your first and last name will be substituted here)

- Drop the **zipped folder** into the dropbox in D2L.
- This must be completed and submitted to the dropbox by the due date/time. All projects turned in after the due date will be penalized 10% each day late up to max three days.
- If you have not completed the project, submit whatever you have. Partial credit is better than no credit at all.