1 Exercises

For the following exercises, pair up with someone else in the class and create a project for each exercise. In your comment section at the top of each of the programs make sure that both of your names are listed in the Author line in the comments at the top of each program.

For each of the following create a new project with an appropriate name and then write a program that solves the given problem. Remember to do Shift+Ctrl+F to format the program, or Shift+Command+F on the Mac to format the code. Also remember the standard comments of at the top, Authors, Date, and Description.

As usual, you will be submitting through MyClasses the java code files and a Microsoft Word docx file, LibreOffice Writer odt, or a text file (which you can create with NotePad++) which contains the following.

- An algorithm for the program.
- Output of at least three runs of each program on different data inputs.

Remember that you can copy and paste output from the Eclipse console area to the word or text program. Only one of you will need to submit the files.

---

1. Write a program that will take the user's name and a statement and output that the person said the statement, as in the following example. The words after the ':' are what the user typed.

```
What is your name? : Don
What do you have to say? : This is a really easy class!

Don said, "This is a really easy class!"
```

2. Write a program that will capitalize a single word that is input from the user. Several test runs are below.

```
Input a single word: tiTLe
Title

Input a single word: JAMES
James

Input a single word: programming
Programming
```

3. Write a program that will generate a number between 1 and 50 inclusively (not printed to the screen). Then have the program ask the user to guess the number. If the guess is correct have the program print that out, if the guess is smaller than the answer have the program tell the user that the guess was too low and then display the correct answer, and finally if the guess was larger then the answer have the program tell the user that the guess was too high and then display the correct answer. Three runs are below.

```
Guess a number from 1 - 50: 25
Your answer is too high, the answer was 20.

Guess a number from 1 - 50: 25
Your answer is too low, the answer was 35.

Guess a number from 1 - 50: 37
Your answer is correct.
```

2 Challenge Exercise

Challenge Exercises are optional, they will be graded as extra credit.

In the last homework exercise set you had the program return the randomly generated card values for two players. The format of the display of each card was a bit unnatural, we would expect something like "3 of Hearts" or the "Ace of Spades". Converting these numbers to the strings Hearts, Spades and Ace are fairly easy to do with conditional statements but we do need them to produce this type of output. A sequence of well chosen string replace commands will do the same trick. Revise the program from the last homework to produce the better output for the cards. **Do not use conditional statements, just string replace commands to do it.** For example, instead of the output,

```
Player 1
Card 1: 13 of 4
Card 2: 10 of 3

Player 2
Card 1: 9 of 3
Card 2: 4 of 1
```

The user would see,

```
Player 1
Card 1: King of Spades
Card 2: 10 of Clubs

Player 2
Card 1: 9 of Clubs
Card 2: 4 of Diamonds
```