CSC 101 Programming Assignment #4 10/24/17

Due date – Thursday November 30th @ 11:59PM

**Program 1:**

This problem is not from the book, but it is based on chapter 8 information. The program will randomly pick a number between one and ten. Then it will ask the user to make a guess as to what the number is. Their inputs should have try and catch blocks to protect from them entering strings or values outside the range, forcing them to reenter. Exceptions should be thrown for all cases, illegal and legal but outside the range. After each guess you will respond with a higher or lower (The direction that they need to go). **If there next guess is not higher or lower, again catch this error and tell them to try again. The range of acceptable inputs changes based on past guesses.** For example, if they guess a 5 and the real answer is higher, any value 5 or below should now throw an exception. Continue until the number is finally guessed. Exceptions should be thrown for every time that the user does not follow instructions.

**Program 2:**

This program is not from the book, but it is based on chapter 9 information. I want you to create a hangman game. The program will start with by asking the user for the hidden word or phrase (can be multiple words). And then ask for how many missed guesses are allowed before you will lose. Then the game will begin, (we are assuming a second person will take over at this point) As the game begins you will show a series of dashes that represent each letter in the word or phrase (blank spaces and punctuation should still show). The user will guess a letter. If that letter exists – all versions of that letter (upper and lower case) will appear among the dashes. If it doesn’t exist – you will be told that you have used up one of your misses and only have X number of guesses left. As you continue to guess letters, correct letters should stay shown on screen. You continue to guess letters until you either unveil all the letters, or you run out of missed guesses. Repeating a letter should count as a miss, but if you have problems with this, leave it as no change to status.

**Program 3:**

This problem is not from the book, but it is based on chapter 10 information. You have N mailboxes in a row. I want you to make this number entered by the user. These mailboxes will be represented in an array. Each mailbox has only two states – open or closed. So make this an array of Boolean, true = closed, false = open. Start with all mailboxes closed. Begin with mailbox #2 (remember this is index 1 in the list) and open it and every 2nd mailbox. Now starting with mailbox 3, and continuing every 3rd mailbox we will open it if it is closed and close it if it is opened. Go to fourth and reverse the status of every fourth mailbox the same way. Continue opening and closing mailboxes every 5th, every 6th, etc… until we get to where we are starting at mailbox N. Report all the mailboxes that are closed after we finish our procedure, do not report the open ones. You should see a recognizable pattern with the few that remain closed.

Each of these programs must be done in a separate file. Name them **Guess.java,** **Hangman.java** and **Mailbox.java**. Name the class in each file these names as well. If you do not name these files correctly you will lose points.

Hand in electronically – (NOT E-mail!!!)

In S-drive CSC 101 folder:

1. Create folder called **projectfour\_firstname\_lastname**
2. Place three files named above in folder. (NOTHING ELSE)