Lab 4

- name your programs exactly as directed, including case!
- autoformat all programs
- wrap long lines
- include your name and a description of the program in comments at the beginning of every program
- choose descriptive variable names

Lab 4

- ID loop patterns
  - when you realize you need a loop, you should think first of the pattern and form as that gives you the framework for the code
  - asked for pattern and form to be identified in comments
- only create one Scanner per program!

Lab 4

- if something in a loop body happens only on the first iteration or the last, consider pulling it out of the loop (before or after)

☹

```
int sum = 0;
for ( int rep = 0 ; rep < 100 ; rep++ ) {
  int roll = (int)(Math.random()*6)+1;
  sum = sum+roll;
  if ( rep == 99 ) { System.out.println(sum); }
}
```

☺

```
int sum = 0;
for ( int rep = 0 ; rep < 100 ; rep++ ) {
  int roll = (int)(Math.random()*6)+1;
  sum = sum+roll;
}

System.out.println(sum);
```

Lab 4 - #1 (Lines)

- error-checking – the number of repetitions should be >= 1
  - the check should be done *before* any lines are printed
  - need to re-read input, not only check whether the value read is valid
  - need a loop in order to keep prompting, reading, and checking until a valid number is obtained
- organization
  - a better strategy is to first get a valid number of repetitions, then print the lines – two separate loops, one after the other

```
// get valid number of repetitions
// print lines
```

☺

```
// repeat
//   prompt and read number of repetitions
//   if number of repetitions >= 1
//     print lines
//     break
//   else
//     print error message
```

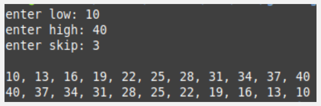
☹

Lab 4 - #2 (Counter)

- don't print , and space after the last thing on the line

```
10, 13, 16, 19, 22, 25, 28
30, 27, 24, 21, 18, 15, 12
```

- be careful not to skip the last thing on the line if it is even multiple



```
enter low: 10
enter high: 40
enter skip: 3

10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40
40, 37, 34, 31, 28, 25, 22, 19, 16, 13, 10
```

- use `System.out.print` for prompts if the user input should appear on the same line

```
enter low: 10
enter high: 30
enter skip: 3
```

Lab 4 - #4 (DiceRoller)

- not scoring correctly
  - not handling 1 at all
  - setting the score to 0 if a 1 was rolled but then continuing to roll
  - computing the sum of the values rolled instead of the max
- score should be the max value rolled, or 0 if a 1 is rolled

Lab 4 - #5 (CrapsOdds)

- The player rolls two dice. The sum of the numbers rolled is called the "initial sum".
- If the initial sum is 7 or 11, the player wins immediately.
- If the initial sum is 2, 3, or 12, the player loses immediately.
- If the initial sum is anything else (4, 5, 6, 8, 9, 10), the player keeps rolling the dice until the sum of the numbers rolled is equal to the initial sum (she wins) or 7 (she loses).

- missing or incorrectly handling the third case (not an immediate win or loss)