# **Lab 4**

Write a method for each of the following

**1. A and B**

Given 2 non-negative ints, a and b, return their sum, so long as the sum has the same number of digits as a. If the sum has more digits than a, just return a without b.

****Test cases****

sumLimit(2, 3) → 5

sumLimit(8, 3) → 8

sumLimit(8, 1) → 9

**2. Squirrel Play**

The squirrels in Palo Alto spend most of the day playing. In particular, they play if the temperature is between 60 and 90 (inclusive). Unless it is summer, then the upper limit is 100 instead of 90. Given an int temperature and a boolean isSummer, return true if the squirrels play and false otherwise.

****Test cases****

squirrelPlay(70, false) → true

squirrelPlay(95, false) → false

squirrelPlay(95, true) → true

**3. Lottery Ticket**

You have a green lottery ticket, with ints a, b, and c on it. If the numbers are all different from each other, the result is 0. If all of the numbers are the same, the result is 20. If two of the numbers are the same, the result is 10.

****Test cases****

greenTicket(1, 2, 3) → 0

greenTicket(2, 2, 2) → 20

greenTicket(1, 1, 2) → 10

**4. Finding ‘xx’**

Count the number of "xx" in the given string. We'll say that overlapping is allowed, so "xxx" contains 2 "xx".

****Test cases****

countXX("abcxx") → 1

countXX("xxx") → 2

countXX("xxxx") → 3

**5. Repeated String**

Given a string and a non-negative int n, return a larger string that is n copies of the original string.

****Test cases****

stringTimes("Hi", 2) → "HiHi"

stringTimes("Hi", 3) → "HiHiHi"

stringTimes("Hi", 1) → "Hi"

**BONUS:**

**Using the principles of Inheritance and/or Encapsulation. Reformat each question into its own separate class, accessed by a MainMenu class**

**Allow the user to choose which class to run on a separate Main method via menu option, allow for user input depending on the choice given.**