**Computer Science 30**   
First Day Class Entry Activity

Use Python code (or pseudo‐code if you were not taught Python in CS20) to answer the following questions, which will be given in **increasing difficulty.** This is designed to get an idea of how well we remember both basic and more involved concepts from CS20.

*Note: This entry activity will not count towards your grade, but will instead give me an idea of each student’s starting ability. Challenge yourself to solve as many of the problems you can in the time allotted.*

**Basics (2)**

*Finish the problem in each “answer” section below. You should be able to copy-paste your work from Thonny for each solution. If you are struggling with these, you should spend time going back to make sure you understand the fundamentals you were taught in CS20*.

|  |  |
| --- | --- |
| Question 1 [sum\_double]  Given two int values, return their sum. Unless the two values are the same, then return double their sum.  sum\_double(1, 2) → 3  sum\_double(3, 2) → 5  sum\_double(2, 2) → 8  **Answer:**  def sum\_double(a , b):  sum = a+b    if a==b:  sum= sum\*2  return sum | Question 2 [sleep\_in]  The parameter weekday is True if it is a weekday, and the parameter vacation is True if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return True if we sleep in.  sleep\_in(False, False) → True  sleep\_in(True, False) → False  sleep\_in(False, True) → True  **Answer:**  def sleep\_in(weekday, vacation):  if not weekday or vacation:  return True  else:  return False |

**More Challenging (2)**

*Once you’ve solved the basics problems above, try these that require more control structures.*

|  |  |
| --- | --- |
| Question 3 [repeat\_front]   Given a string and a non-negative int n, we'll say that the front of the string is the first 3 chars, or whatever is there if the string is less than length 3. Return n copies of the front;  repeat\_front ('Chocolate', 2) → 'ChoCho' repeat\_front ('Chocolate', 3) → 'ChoChoCho' repeat\_front ('Abc', 3) → 'AbcAbcAbc'  **Answer:**  def repeat\_front (str, n):  return str[0:3]\*n | Question 4 [array\_look]  Given an array of ints, return True if the sequence of numbers 1, 2, 3 appears in the array somewhere.  array\_look ([1, 1, 2, 3, 1]) → True array\_look([1, 1, 2, 4, 1]) → False array\_look([1, 1, 2, 1, 2, 3]) → True  **Answer:**  def array\_look(nums): |

**Max Difficulty (1)**

*Well, not really. But the hardest of this set, at least!*

|  |
| --- |
| Question 5 [text\_match]   Given 2 strings, a and b, return the number of the positions where they contain the same length 2 substring. So "xxcaazz" and "xxbaaz" yields 3, since the "xx", "aa", and "az" substrings appear in the same place in both strings.  text\_match('xxcaazz', 'xxbaaz') → 3 text\_match('abc', 'abc') → 2 text\_match('abc', 'axc') → 0  **Answer:**  def text\_match (a, b): |