Name: _____ **Section:** 1 2 3 4

1 = Stouder, 2nd–3rd periods. **2 =** Stouder, 4th–5th periods. **3 =** Mutchler, 7th-8th periods. **4 =** Rupakheti, 9th-10th periods.

Use this quiz to help make sure you understand the videos/reading. ***Answer all questions.*** Make additional notes as desired. ***Not sure of an answer?*** Ask your instructor to explain in class and revise as needed then. ***Please print two-sided if practical.***

Throughout, where you are asked to "circle your choice", you can circle or underline it (whichever you prefer).

From the Course Home Page: ***Test 1***

1. Regarding Test 1: What are the ***day of the week and date*** of that test?

   _____

   The ***time***? _____ The room for your section? _____

Video: ***The Accumulator Pattern – Part 2*** [9:36 minutes]

2. What is the value of **x** after the following code executes: ___

```
x = 2
x = x + 1
x = x + 2
x = x + 3
```

3. What is the value of **y** after the following code executes: ___

```
y = 2
y = y * 2
y = y * 3
```

4. What are the values of **x, y** and **z** after the following code executes:

```
x = 1
y = 2
z = x + y

x = x + 2
y = y * 2
z = z + x + y

x = x + 3
y = y * 3
z = z + x + y
```

x is _____ **y is** _____ **z is** _____

5. Implement the following function (here, on paper – it is fine to make small errors):

```
def add_them(n):
    """ Returns 2 + 4 + 6 + 8 + 10 + ... + 2n """
```

6. Implement the following function (here, on paper – it is fine to make small errors):

```
def multiply_them(n):
    """ Returns 2 * 4 * 6 * 8 * 10 * ... * 2n """
```

7. If you are doing a ***counting*** problem, what is a good choice for the name of the variable that does the counting: _____
(Hint: **x** or non-descriptive names like it are poor choices.)

8. If you are doing a ***summing*** problem, what is a good choice for the name of the variable that does the summing: _____
(Hint: **x** or non-descriptive names like it are poor choices. Also, **sum** is a poor choice because it happens to be the name of a built-in function.)

9. Suppose you wanted to count the number of integers from 3 to 40,000 that are prime. What **for ... in range ...** statement would you use? You do NOT have to provide the body of the *for* loop. (Hint: There are at least two natural approaches to solving this problem; one of those runs twice as fast as the other.)

10. Implement the following function (here, on paper – it is fine to make small errors):

```
def count_them(m, n):
    """
    Returns the number of integers from m to n, inclusive,
    whose cosine is less than 0.5.
    Preconditions: m and n are integers with n >= m.
    """
```

Textbook Reading: ***Sections 3.3 – Nested Branches*** (pages 106 - 109) and ***3.4 – Multiple Alternatives*** (pages 109 - 112)

11. In a game program, the scores of players A and B are stored in variables **scoreA** and **scoreB**. Assuming that the player with the larger score wins, write an **if/elif** sequence that prints out ***'A won'*** or ***'B won'*** or ***'Game tied'***.

12. Beginners sometimes write statements such as the following:

```
if price > 100:
    discounted_price = price - 20
elif price <= 100:
    discounted_price = price - 10
```

Explain how this code can be improved.

13. Suppose that the following (***very silly!***) situation were true:
    - For any boy whose age is greater than 18, his favorite ice cream flavor is chocolate.
    - For any other boy, his favorite ice cream flavor is pistachio.
    - For any girl whose age is greater than 50, her favorite ice cream flavor is mint chocolate chip.
    - For any girl whose age is between 16 and 50 (including both 16 and 50), her favorite ice cream flavor is banana.
    - For any other girl, her favorite ice cream flavor is Oreo.

    Suppose that for person, the variable **`gender`** has been set to either **`'boy'`** or **`'girl'`** (to indicate the person's gender) and the variable **`age`** has been set to the person's age. Write a ***nested*** set of **`IF`** statements that print this person's favorite ice cream flavor.

14. It so happens that you can use the word **`and`** inside IF and ELIF statements, like this:

```
if x > 12 and x < 47:
    ...
```

    You'll read about ***Boolean operators*** like **`and`** in a forthcoming reading. Today, try this: using the **`and`** operator as you would guess, repeat the previous problem, but solve it using ***multiple alternatives*** (**`if/elif`**) INSTEAD of a ***nested*** set of IF statements. Which solution is better, do you think? (**Note:** It is OK if you don't understand how to do this problem; we will go over it in class. But give it a try.)

15. **Think about it:** (you do not have to turn it in, answers are in the book, so you can check your answers after you think about them): Self-check #28 and 30 on page 117.