[answers in web view]

Name: ______________________________
Student ID: ______________________________

Total points: 70

1. Name the five steps to **top-down** problem solving as described in the text.
   (It's okay if you can't remember the exact wording; the concepts are more important.)
   Also: describe each step in your own words, and why it is important! **[5]**

   -
   -
   -
   -
   -

2. Name the five program **control/flow** abstractions we talked about in class. For each one, name a Python language construct which implements the abstraction. (There is one abstraction which we are not covering in Python; you may write "N/A" for the Python for that one.) **[5]**

   -
   -
   -
   -
   -

3. Write "yes" or "no" next to each of the following six strings to indicate whether it can be used as a valid name for a user-defined Python variable. If "no", indicate why. **[5]**

   `python`
   `in`
   `If`
   `my.Apples`
   `____` *(4 underscores)*

4. Use the Python `range()` function to create the following list.
   (There are several possible correct answers.)
   (Don't worry about Py3's list/iterator issue; just use range().) **[2]**
   `[7, 3, -1, -5]`

19. Describe the **bug** in this Python loop, and fix it: [3]

```
countdown = 10          # Count from 10 down to 1

while countdown > 0:

        print(countdown, end=' ')

        countdown - 1

print("Blast off!!")
```

20. The following is meant to be an interactive gradebook that converts from a percentage into a letter grade: 95 and above is A+, 85-95 is A, 80-85 is A-, and everything below is fail (it's a really tough program!). The program should prompt the user 100 times for grades. Unfortunately, the program was written by a terrible coder (me!), so there are numerous **bugs** in the code. Find all the bugs and fix them. [8]

```
For i : range(1 .. 100)

        grade == input('What\'s the grade percentage? ')

        if grade > 95

                print("That's an A+!')

        if 85 > grade > 95:

                print("""An A is really good!""")

        elsif 80 < grade:

                print( You got an A-! )

        else grade < 80:

        print("Sorry, you need an A- to pass!")
```

21. **Evaluate** each of the following Python expressions exactly as given, or if it produces an error, describe the error. Assume each expression is independent of the others, and is the only command in a new Python session. **[15]**

    a. `list(range(2))`
    b. `list(range(2,-2))`
    c. `list(range(2,-2,-2))`
    d. `5 % 13`
    e. `3 ** 2 < 8 and 5 % 0 == 0`
    f. `2 - 2 * - 2 ** 2`
    g. `0.8 + 0.1 + 0.1 = 1.0`
    h. `'7' + '3'`
    i. `"%04.1f" % 2`
    j. `"My name is %s" % Joe`
    k. `"Name: %s, ID: %04d" % 'Bob', 173`
    l. `"03.0f" % 2`
    m. `ord('E') - ord('B')`
    n. `'E' - 'B'`
    o. `'pi' + 2*'z' + 'a'`

22. Describe and contrast the roles of the **producer** and the **director** in a team. Describe specific examples of what these roles might look like for a particular kind of task, e.g., developing software, building a bridge, making a movie, etc. **[6]**

23. Compare and contrast **static** typing versus **dynamic** typing, including pros/cons. **[4]**

24. Describe and contrast the **requirements** document and the **design specification** for a software project. **[5]**

25. **Newton's method** for computing square roots is an iterative method which starts from an initial guess of the square root and gets closer to the true square root with each step, according to the following algorithm:

    Let x be the number we wish to take the square root of.
    Let r be our current guess for the value of the square root.
    Calculate our next guess via (r + x/r)/2.
    Repeat until the guesses don't change by more than, say, 0.0001 (our desired level of precision).

    Your task is to code a Python **function**, called `newton(x)`, that uses Newton's method to calculate and return the square root of x.

    a. Write a **docstring** for this function, including detailed pre-condition and post-condition. **[4]**

    b. Write **Python code** implementing this function. Use x/2 as the initial guess for the square root. Obviously, you may not use math.sqrt for this! (It turns out that math.sqrt does use a similar algorithm -- Newton's method converges to the right answer quite quickly!) **[8]**

    Comments and pseudocode are not required but may earn you partial credit if they show good design thinking. Little credit will be given for uncommented incorrect code. Note that your code is not required to print() anything on the screen or to input() anything from the console.