[answers in web view]

Name:_____
Student ID:_____

Total points: 70

1. Fill in the blank: [6]

   (a) In software development, the document that describes the functionality you intend to deliver to the client is called the _____.

   (b) When we wish to `print()` an object to the console, which method of the object does Python invoke? _____.

   (c) In a Python class, the **`__init__()`** method is the special method called the _____.

   (d) The Python method on file objects that sets the **file pointer** (location within the file) is called _____

   (e) What English part of speech should a **method** name be? _____

   (f) Let `myList = [ 9, 8, 7, 6 ]`. What is `myList[1:3]`? _____

2. What are **exceptions** used for? What happens when one is raised? [4]

3. Compare and contrast Python **lists** with C/M2 arrays [4]

4. What is output by the following block of Python code? [3]

```
myList = [ "Hey bub!", ("hey", "bub!"), range(10, 40, 5), [ range(5) * 2 ] ]
for item in myList:
        print( len(item), end=' ' )             # print all on one line
```

5. What is **serialization**, and why is it useful? What does it look like in Python? (Just describe the library and primary functions; no need for complete code.) [5]

6. Describe all the various **file modes** in which Python can open a file, as though you were writing a manual for the function `open()`. Indicate any errors that might occur and anything that a Python programmer using `open()` might need to watch out for. [5]

7. What is a **stub function**, and why might such things be useful in software development? Describe an example.[5]

8. Recall that in Python int/float/bool parameters are essentially passed by **value**, and lists/object parameters are essentially passed by **reference**. What is printed by the following block of Python code? [4]

```
def double_me(x):
        x *= 2
y = 3
z = list(range(3))
double_me(y)
double_me(z)
print( 'y =', y, ', z =', z )
```

9. Define or describe each of the following terms in object-oriented programming: [5]

   (a) Class:

   (b) Instance:

   (c) Attribute:

   (d) Method:

   (e) Constructor/initializer:

10. What are **set/get** (mutator/accessor) methods, and why are they a good idea? Describe an example.[5]

11. Design a **class** `Song` that represents a single song (e.g., MP3) in your music collection (e.g., iTunes). What **attributes** are needed (and what types are they)? What **methods** are desirable? No Python code is needed, just discussion. **[6]**

12. Write a Python function `clicker(x,y,s)` (taking 3 parameters) which uses Zelle's `graphics.py` library to draw a window with a **button** in it, and waits for the user to click. The button should be **centred** around the coordinates (x,y) and have a text string **label** given by the parameter `s`. When the user clicks, the function should **return** a boolean value indicating whether the user clicked the button or not. **[8]**

    Docstring, pseudocode, comments, etc. are not required, but may be helpful for partial credit if your code isn't right. You may assume `graphics` has been imported, but do not assume a window has been made already. It's okay if you don't remember the exact names of the classes/functions in the `graphics.py` library; just use pseudocode or make a guess as to the name. Don't worry about whether the label will fit within the button; just choose some reasonable dimensions for the window and the button.

13. Write a Python program to read in integers from the file "input.txt", one per line, square each number (i.e., multiply it by itself), and print the results (one per line) to the file "output.txt". For example, "input.txt" might look like the following:

    ```
    13
    -5
    0
    100
    ```

    Lines in the input file which do not conform to this format should be skipped (don't assume that the input file will be formatted properly!). Your program would output something like this:

    ```
    169
    25
    0
    10000
    ```

    You should also ensure your program doesn't crash if the input file is not found, or if file permissions are incorrect, etc. Docstring, pseudocode, comments, etc. are not required, but may be helpful for partial credit if your code isn't right. **[10]**