

Total points: 100

1. Fill in the blank: **[11]**

- (a) "**Computers** are _____, and computing scientists are _____."
- (b) In software development, the document that describes what the client desires for the project is called the _____.
- (c) When a new instance (object) is created in Python, the method named _____ is invoked.
- (d) Testing of a single component in isolation (as opposed to integration testing) is called _____ testing.
- (e) What word describes functions which **invoke** themselves? _____
- (f) The Python method which **serializes** an object and writes it out to a file is in the _____ library and is called _____.
- (g) What English part of speech should an **attribute** name be? _____
- (h) Let `myList = [1, 3, 5, 7]`. What is `myList[2:2]`? _____
- (i) What method returns a list of all the **indices** in a **dictionary**? _____

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

3. Compare and contrast **static** typing versus **dynamic** typing. What are pros/cons of each?
[4]

4. What is **wrong** with the following block of Python code? Indicate how you would fix the bug. [4]

```
def fun(fu):  
    fa = fu * fun(fu - 1)  
    return fa  
print( fun(5) )
```

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

6. Consider the concepts of **call-by-value** and **call-by-reference**.

(a) Define and contrast these two terms as general **concepts** (i.e., independent of Python). [4]

(b) Write **Python** code which clearly demonstrates the difference between the two. [4]

7. **Evaluate** each of the following Python expressions exactly as given. If it produces an error/exception, describe the error in words. Assume each expression is independent of the others. For all expressions, assume only the following initialization: **[15]**

```
myPear = "Red Bartlett"  
from math import pi
```

- (a) `19492016 % 2`
- (b) `[2, 4, 6] * 2`
- (c) `[2, 4, 6] + 8`
- (d) `2 ** 3 > 5 or 3 / 0`
- (e) `int("2 pears")`
- (f) `"I have %06.2f pears" % 3.1415`
- (g) `"03d%s" % (15, 'pears')`
- (h) `'b' in myPear`
- (i) `Bartlett.upper()`
- (j) `range(myPear)`
- (k) `myPear[2:6]`
- (l) `chr(ord('P') + 3)`
- (m) `'pi' + 'e'`
- (n) `'pie' - 'e'`
- (o) `pi + 'e'`

8. Define or describe each of the following terms in object-oriented programming: **[6]**

- (a) Class:
- (b) Instance:
- (c) Attribute:
- (d) Method:
- (e) Constructor/initializer:
- (f) Overloading:

9. Name the five steps to **top-down** problem solving as we discussed in class. (It's okay if you can't remember the exact wording; the concepts are more important.) **[3]**

10. Discuss the limitations of the software development model used in the previous question, and describe at least two alternate models that we talked about in class that might better reflect real-world software development. Discuss pros and cons of the alternate models as compared with the model used in the previous question. [8]

11. Consider a **class** `Person` that represents a single person in a social network like Facebook.

(a) What **attributes** are needed (indicate type of each attribute)? What **methods** are desirable? No Python code is needed, just discussion. [6]

(b) What **other classes** might you need to design a social network platform? What **relationships** might exist between those classes and the `Person` class? Discuss and sketch a **class diagram**. [6]

12. Write a Python program which prints (on the screen) every word which appears exactly **once** (and no more than once) in the file "input.txt".

Ignore case. Words are separated by spaces or newlines, and all punctuation, digits, etc. can be treated the same as letters. For instance, if the file "input.txt" contains:

Hello, this is a text file. hello, here's a TEXT file.

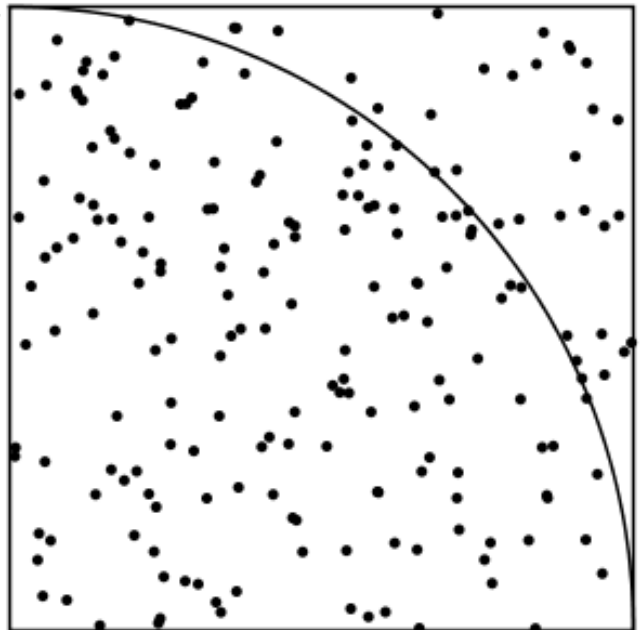
then your program should output "this here's is".
(Order of the words in the output does not matter.)

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]**

13. One (very slow) way to calculate the value of π is to compare the area of the unit quarter-circle with the area of the unit square.

We can estimate this by generating random points in the unit square (akin to throwing darts at the unit square), and counting how many lie within the quarter-circle. The ratio should approximate $\pi/4$ as the number of points increases. Some helpful hints:

- The function `random()` in the library `random` returns a float between 0 and 1 (including 0.0 but not 1.0).
- Pythagoras' theorem states that for any right triangle with legs a and b and hypotenuse c , $a^2 + b^2 = c^2$.



[Image courtesy [Rochester Institute of Technology](#)]

(a) Your task is to design a function `circle_pi()` that uses the above algorithm to estimate π . The caller should be able to specify the number of points (darts) to use in the approximation. The function should be modular so that other code can easily use it -- it should **not** do any console I/O! Write a comprehensive **docstring** for this function (but don't implement it yet!), including pre- and post-conditions. **[4]**

(b) Write a comprehensive **test suite** for this function, including both valid and invalid input. Clearly indicate the invocation and the expected result. **[4]**

(c) **Implement** this function in Python. Pseudocode, comments, etc. are not required, but may be helpful for partial credit. You should **not** need to import `math`! **[6]**