1. Programming languages

- Python and Java use different typing systems. What kind of typing system does each language have? Explain in 1 to 2 sentences the characteristics of each language's typing system. (You may choose another language than Java if you wish).

2. Data primitives

- Python and other languages store strings internally as character arrays. Assuming the characters are all 8-bit ASCII, what are the necessary (minimum) number of bytes required to store the string,

  "It was the best of times, it was the worst of times."

- A physicist creating a computer simulation needs to store the position of 1000 particles in space. For this purpose, he needs just 3 doubles for each particle, one for each the x, y, and z coordinates. How many bytes are necessary to store this information?
- If the beginning of an integer array is at memory address 0x0002 (in hex) and you want to the value of the 3rd element in the array, from what memory address do you begin and end reading from to get the data at that position in the array?

3. Data Types

- ***Abstraction*** is a cross-cutting concept in computer science and the programming discipline. What are at least two practical reasons abstraction is used when solving problems?
- Is there a difference between the concept of Abstract Data Type (ADT) and a particular class definition in Python? Use "computer science" terms you heard in class and give examples.

4. Pepperoni Pizza Factory

- You must create two classes, a *Pizza* and a *PizzaFactory*. A PizzaFactory is an iterable datatype that will take one unit each of cheese, dough, tomato sauce, and pepperoni, then instantiate and return a new Pizza. For keeping track of the ingredients, the PizzaFactory has a counter stored internally

for the amount of each of the ingredients the factory currently has in its inventory. If the factory is out of ingredients when it tries to make a pizza, it should raise a StopIteration exception. You should create a method for each of the ingredients to restock the inventory.

5. Linked Structures

- Many times in class we studied and showed examples of how a linked list queue works by popping from the right, and pushing from the left. For this implementation, you will instead pop from the left, and push from the right. Write the rest of the code from the following template. Note you cannot use a tailnode reference for this implementation. Instead, you must traverse the list to get to the end.

```
class Node (object):

    def __init__(self, value):
        self.nextNode = None
        self.prevNode = None
        self.value = value

class Queue (object):

    def __init__(self):
        self.headNode = None

    def push(self):
        # write your code

    def pop(self):
        # write your code
```

6. Generator and Iterators

- Create a generator (by writing a function that has a yield in it) that yields the number of bacteria in a colony at each time step. The next number in the sequence should contain 20% more bacteria than the previous time period. Your sequence should start with the number entered as the argument.

$$S_n = S_{n-1} * 1.2$$

7. Complexity

- Time complexity is an ever-increasing hierarchy of various growth functions. List as many growth functions as you can. For each grouping of growth functions, annotate whether that function is in the constant, linear, or exponential.
- For this problem, consider queues and arrays. *Queue A* is a linked list with only a head node and the nodes are singly linked to its next node. *Queue B* has both a head node and tail node and is doubly linked. The array implementation of a queue is a contiguous block of memory and assume you store the length of the array in a variable. Do not worry about the time cost of resizing the array. What the time complexity expressed in Big O notation for the following operations:

    1. Push into Queue A
    2. Push into Queue B
    3. Push into Array from the beginning
    4. Pop element from Queue A
    5. Pop element from Array
    6. Search by value for element in Queue A
    7. Search by value for element in Array
    8. Index by position the value of an element in Queue A
    9. Index by position the value of an element in Array
    10. For each value in the Array, find if there is a duplicate in the Array

Consider the following functions:

```
def A(n):
    while n > 1:
        n = n // 2

def B(n):
    for i in range(n):
        A(n)

def C(n):
    for i in range(500000000000):
        B(n)
```

- What is the time complexity of running A(n)?
- What is the time complexity of running B(n)?
- What is the time complexity of running C(n)?

8. Functional Programming

Python has higher-level functions such as map, filter, and reduce. In the problems, implement in code solutions without any control structures such as (while loops, for loops, etc). You may use len() and math.sqrt() in your solution.

- Compute the mean using only a map and a reduce. Hint- you can compute the mean by mapping the reciprocal of the number of elements across the data set and then using a reduce to sum the result. Write this in code.
- Compute the standard deviation. The standard deviation is often represented as the square root of the average of the squared differences of values from their average value. Hint- Map the difference from the mean as a function across the elements. Map the square function across the elements. Reduce the result by summing everything. Divide by the number of elements, take the square root.