NAME _____ **ICA-8**

NetID_____ @email.arizona.edu

Work with your neighbor.

This section involves working on complexity and exceptions.

Problem 1. Complexity

a) What is the worst-case big-O complexity of the following code fragment?

```
n = int( input() )
for i in range(n):
        x = x + 1
```

b) What is the worst-case big-O complexity of the following function?

```
def fun2(n):
    k = n*n
    sum = 0
    for i in range(k):
        sum += i
    return sum
```

c) What is the worst-case big-O complexity of the following function?

```
def print_sums(numlist):
   m = 100
   for x in numlist1:
      for y in range(m):
          print(x + y)
```

d) Write a snippet of code that is $O(n^3)$.

Problem 2. In this problem, you will write two different versions of a function and determine their run-time complexity. Write a function `has_dups(alist)` that takes a list of integers and returns `True` if `alist` contains duplicate values and `False` otherwise. If `alist` is empty, the function returns `False`.

a) In the first version, use nested loops. What is the complexity of your function?

b) In the second version, use a dictionary to keep track of whether a value has been seen before. Once a value has been seen, the function can immediately return.

c) Since the function only iterates through `alist` once (worst case), its complexity appears to be O(n). What do we need do know about dictionary operations in order to give a thorough answer?

Problem 3. Explain why the following code demonstrates a poor use of exceptions.

```
try:
    infile = open(filename)
    z = 1/n
except:
    print("ERROR: could not read file " + filename)
```

Problem 4. What's the time complexity of `insert()` and `append()` methods in Python List? Briefly explain why.