CS-160 **Stacks, Queues, and Analysis** **2016-03-15**

This Exam is being given under the guidelines of the **Honor Code**. You are expected to respect those guidelines and to report those who do not. Answer the questions in the spaces provided. If you run out of room for an answer, continue on the back of the page. There are 6 questions for a total of 53 points.

Name: _____

1. (5 points) Rank the following Big-O classes in order from 1 to 5 where 1 is the most efficient (fastest) and 5 is the least efficient (slowest).

   - $O(N)$
   - $O(Nlog(N))$
   - $O(1)$
   - $O(log(N))$
   - $O(N^2)$

2. (5 points) Draw the contents of the stack after the following operations

```
myStack = Stack()
myStack.push(1)
myStack.push(2)
myStack.pop()
myStack.push(3)
myStack.peek()
myStack.push(4)
myStack.pop()
```

3. (5 points) given the following code, what is the output?

```
1   class Animal:
2       def __init__(self,name):
3           self._name = name
4       def speak(self):
5           print("Hello")
6       def get_name(self):
7           return "My name is {}".format(self._name)
8
9   class Dog(Animal):
10      def __init__(self,name):
11          super().__init__(name)
12          self.numLegs = 4
13
14      def speak(self):
15          print("Woof Woof")
16
17  class Cat(Animal):
18      def get_name(self):
19          return "My Cat name is {}".format(self._name)
20
21  spot = Dog("Spot")
22  boots = Cat("Felix")
23  spot.speak()
24  boots.speak()
25  print(spot.get_name())
26  print(boots.get_name())
```

4. Given the following code for a Node class and a Stack class.

```
1  class Node:
2      def __init__(self,initdata):
3          self.data = initdata
4          self.next = None
5
6  class Stack:
7      def __init__(self):
8          self.size = 0
9          self.top = None
10
11     def push(self,item):
12         if self.top == None:
13             self.top = Node(item)
14         else:
15             newNode = Node(item)
16             newNode.next = self.top
17             self.top = newNode
18         self.size += 1
19
20 myStack = Stack()
```

(a) (5 points) Draw a picture to represent the internals of the stack after calling: `myStack.push(3)`, `myStack.push(7)`, `myStack.push(1)` in that order.

(b) (5 points) Write a pop method for the stack. Make sure that you take into consideration all edge cases.

question Using the following code fragment:

```
1   sum = 0
2   for i in range(n):
3       for j in range(n):
4           sum = sum + 1
5   for p in range(0,n*n):
6       for q in range(p):
7           sum = sum - 1
8   sum = sum + 10
9   sum = sum + n
10  sum = sum - 10 * n
```

(a) (2 points) Using Big-O notation, What is the worst case performance for lines 2–4?

(b) (2 points) Using Big-O notation, What is the worst case performance for lines 5–7?

(c) (2 points) Using Big-O notation, What is the worst case performance for the lines 8–10?

(d) (2 points) Using Big-O notation, What is the overall worst case performance?

5. (10 points) A Palindrome is a word that is spelled the same both forward and backward. For example, 'radar' is a palendrome, as is the phrase 'a toyatas a toyota'. Write a function that that takes a word or phrase (with the spaces removed) and returns True if the word is a palendrome and False otherwise. A Stack must be an integral part of your solution which will be $O(n)$.

6. (10 points) Implement a Queue using **two Stacks**. Implement the methods `__init__`, `enqueue`, `dequeue`, and `size`.