

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 8 questions for a total of 100 points.

Name: \_\_\_\_\_

1. (5 points) Convert the following expression to its fully parenthesized form.

$$A - B * C / D + E + F$$

2. (5 points) Convert the following expression to its Postfix form.

$$A / B * C + D$$

3. (5 points) Show how the stack is used in evaluating the following postfix expression.

$$7\ 13\ *\ 12\ 8\ -\ 4\ +\ 10\ *$$

4. (5 points) Convert the following expression to prefix notation.

$$A * B / (C - D) + (E * F)$$

5. (20 points) For this question you must implement a Queue class. However you may not use a list in your implementation. You **must** use two stacks. Implement the methods `__init__`, `enqueue`, `dequeue`, and `isempty`.

6. Write a recursive function that takes a string as a parameter and returns the reversed string as the result.

(a) (5 points) What is the base case?

(b) (5 points) What is your ‘recursive step?’

(c) (10 points) Write the function `reverse(s)`

7. Suppose you are writing a banking application. In your application you have several different kinds of accounts. For any kind of account you can deposit money, withdraw money, and check the balance. Checking Accounts do not accrue any interest, give ten free transactions per month, and charge 25 cents for each transaction after the tenth. A savings account earns interest that compounds monthly based on the balance on the last day of the month.
- (a) (10 points) Draw a picture to indicate the inheritance hierarchy you will use in this application. Indicate on the picture where the instance variables belong.
- (b) (10 points) Implement the classes on your inheritance hierarchy. In particular you should write the constructor, and the methods needed to allow for correct deposits and withdrawals for every account.

8. (20 points) Given the following mystery function. Trace the calls and show what it prints out when called as shown.

```
1 def mystery(n,A):
2     for i in range(2):
3         if n < 2:
4             A[n] = i
5             mystery(n+1,A)
6         else:
7             A[n] = i
8             print A[0],A[1],A[2]
9 B = [0,0,0]
10 mystery(0,B)
```