CSI 33 Midterm Exam Sample

A sample Midterm Exam with additional questions

True/False and Multiple Choice Quiestions

(a) Class **variables can be shared by all instances of the class**

True or False?

True/False and Multiple Choice Quiestions

(b) Which of the following is a Θ(n) operation?

I. Sorting **a list with Selection sort**

II. **Finding the i^{th} item in a Python list.**

III. **Re-assigning the element at the end of a Python list.**

IV. **Deleting an item from the middle of a Python list.**

True/False and Multiple Choice Quiestions

(c) Which of the following is not true of Python dictionaries?

I. They are implemented as hash tables.

II. Lookup is very efficient.

III. Values must be immutable.

IV. All of the above are true.

True/False and Multiple Choice Quiestions

(d) By definition, a queue must be a(n):

I. array-based structure
II. linked structure
III. FIFO structure
IV. LIFO structure

True/False and Multiple Choice Quiestions

(e) A queue allows for the inspection of items at either end

True or False?

Time efficiency questions

1. Give a theta analysis of the time efficiency of the following code fragment. Explain.

```
n = int(input("Enter a positive integer:"))
l = []
while n > 1:
    l.append(n)
    n = n/3
```

Time efficiency questions

2. Assuming that we keep the references to the **head** of the linked list and to the **last element** (**tail**) of the linked list and the user provides a valid input (a positive integer), give a **Theta analysis** of the *time efficiency* **T(n)** of the program. Justify your answer.

```
n = int(input("Enter a positive integer:"))
if n > 0:
   myList = LList()
   for i in range(n):  myList.append(randint(0,1000))
   for i in range(n):
      tmp = myList.pop()
      myList.insert(0,tmp)
else: print(n,"is not a positive integer")
```

Queue, Stack, etc. questions

1. The integers 2, 17, 2, 34, 12 and 9 are inserted into the Queue in the given order.

Then three elements are dequeued, and the following elements are enqueued: 8, 7, 5, and 1 in the given order.

Give the order in which all the values will be retrieved from the queue.

Queue, Stack, etc. questions

2. Given two sorted stacks, st1 and st2, in an order such that the smallest element is at the top of the stack and the greatest element is at the bottom of the stack, create a new stack st3, which contains elements from both stacks, st1 and st2, and preserves the order of the original stacks (the smallest element is at the top and the greatest is at the bottom). You can only use Stack ADT operations, you cannot access the underlying representation of the Stack.

Queue, Stack, etc. questions

3. Show pictorially how the following postfix expression can be evaluated using Stack container:

1 2 3 * + 6 2 / -

Infix, potfix and prefix expressions

4. Evaluate the valid **prefix** and **postfix** expressions:

(a) 1 2 3 * + 6 2 / -

(b) ÷ √ ^ 3 2 ^ 4 2 + - 10 8 3

Infix, potfix and prefix expressions

5. Re-write the algebraic expression in infix notation to its prefix and postfix notations.

$$\left(\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\frac{4}{5}+\frac{5}{6}\right)^4$$

Python memory representation

1. Give pictorial representation of the Python's memory during execution of the following code. Show the result of `print` statements.

```
def func(a,b,c):
    a.append(c)
    b = b + ", world!"
    c = c/5
    a = [1,2,3]
    print(a,b,c)
def main():
    l = ['a','b','c']
    d = "Hello"
    k = 25
    func(l,d,k)
    print(l,d,k)
```

Recursion

1. For the given function definition:

```
def function(items):
    '''pre: items is a list of integers '''

    if items == []: return 0

    else:
      return items[0] + function(items[1:])
```

(a) Trace the call of `function([5,8,9])` (show the recursive calls and return values).

(b) Figure exactly how many additions does it do.

Coding questions

1. Write a program to evaluate a valid prefix expression (you can use your homework, where you wrote the code for postfix expression evaluation).

For testing you can use the following expressions:

1) + * 1 2 3 = 1*2+3 = 5
2) - 5 + * 3 4 5 = 5- (3*4+5) = 5 - 17 = -12
3) * + 2 3 4 = (2+3)*4

Coding questions

2. Write definitions of two new abstract data types (ADTs): Patron and Library.

Assume that we want to create a *simple library*:

- it has *a list of different book titles* (assume we only record the title of the book),
- a *list of patrons*,
- a *library name*, and
- a *library id*.

Each *patron* has

- a *name*,
- an *id*, and
- the *list of books on loan* (these can also be book titles).

Feel free to add more attributes as you see fit.

Coding questions

2. continues

Each Patron instance should be able to:

- return the patron's name and id,
- return the alphabetized list of book titles the patron borrowed and did not return yet,
- return the number of books on loan,
- add a book title to the list of books on loan,
- return a book, i.e. remove a book title from the list of books on loan.