Specification for Final Exam Review: Practice Problems

This is not a Tutorial. There are no marks associated with this set of questions - they are purely for practice and to help you prepare for the final exam. Please note that this set of problems are not necessarily comprehensive with regards to all the topics covered in this course, though it will attempt to cover some points that students typically had the most issues with during the course of this semester's tutorials, as well as some topics that we weren't able to cover in these tutorials. You are not obligated to complete this set of questions, nor will an answer sheet be posted.

Topics Covered

Variables and Data Types

Input Statements

Given the following snippet of code, answer the following questions:

1. What is the result of the first print statement?

2. What is the intended result of the second print statement? Is it working as intended?
3. If your answer for **2.** was No, then how would you fix it?

```
inputVar = input("What is your age?")

print(inputVar)
print(inputVar*2)
```

Operations on Variables

The print statement in Python is used to output text to the console so that a user can see it. It is also a very powerful tool for debugging, provided you format and use them properly. Given the following sets of print statements, answer the following:

1. What is the output, if any, of each of the print statements?
2. For the one(s), if any, that fail to output, how can they be fixed? What was wrong with them?

```
strVar1 = "Hello world"
strVar2 = "I like apples"
intVar = 21
boolVar = False

print(strVar1 + strVar2)
print(strVar2 + intVar)
print(strVar2 + strVar1, boolVar)
```

Logical Operators and Control Structures

Tracing the Output

What is the final output of the following snippet of code?

```
a = 25
b = False
if a < 25 or not b:
    b = not b
elif a >= 25 and b:
    a /= 5
else:
    a += 7
    b = False
print(a, end=": ")
print(b)
```

Verifying Code

The following piece of code is designed to ask the user for some input and, depending on the response, return something. Unfortunately, it's not working in its current state. How do you propose to fix it? Are there any other issues that don't prevent the code from running, but might be an efficiency concern?

```
a = input("Give me a number between 1 and 3 inclusively")
if a = 1:
    print("You selected 1!")
elif a = 2:
```

```
    print("You selected 2!")
elif a = 3:
    print("You selected 3!")
elif a > 3 or a < 1:
    print("You entered an invalid value!")
```

Looping Control Structures

Fibonacci Sequence

Given the following piece of code that computes and prints out the n-th value of the Fibonacci Sequence, add some code in the looping structure to complete it as intended. The area you need to add is marked as '**Add code here**'.

```
val1 = 0
val2 = 1
n = int(input("What n-th value of the Fibonacci sequence do you want to know?"))
for i in range(n-1):
    # Add code here
return val1
```

Different Types of Looping Control Structures

There are two major types of looping control structures in Python: the **While** loop and the **For** loop. When should each be used?

Printing a Grid

The following code is intended to be used to print out a 2D list as a nicely formatted grid. Unfortunately, the output isn't quite right. What would you modify/add to fix this code?

```
def printGrid(aList):
    for i in range(len(aList)):
        for aList in range(len(aList)):
            print(j, end=" ")
        print()
```

Functions and Function Scope

Function Variables/Scope

Is the following set of function definitions okay? If not, explain why.

```
def func1(variable1):
    variable1 = 5
    ...
    return variable1
def func2(variable1):
    variable1 = 2
    ...
    return variable1
```

Writing a Square Root Function

Write a function that returns the square root of a given parameter. You are not permitted to use the sqrt function. Your function should function for float values as well, both as inputs and outputs.

Linear Collections

Accessing Values in a N-Dimensional List

What is are the values being printed out with the following snippet of code? If the print statement results in an error, simply write ERROR for that line, but assume that the remaining code will still run.

```
lst = [[1, 2, [5, 2], 1, "hello"], False, 2, [1, 5, 6], [True, "Hello"]]
print(lst[0][2][1])
print(lst[0][0][1])
print(lst[0][4][3])
print(lst[4][0])
```

Duplicating 2D List Values

Write a function that takes in a single 2D list as input and returns a 2D list as output. This function must double each value in the 2D list, wherein strings, ints, and floats must be accounted for. Do not worry about boolean values for this problem. You may not use list comprehension for this problem.

Generating an Initial-State Chessboard

You must write a function that generates a 2D list representing the initial state of a chessboard. The size of the grid is 8*8, with the following representations:

(a) -'s for blank spaces

(b) p/P's for pawns

(c) r/R's for rooks

(d) n/N's for knights

(e) b/B's for bishops

(f) q/Q's for queens

(g) k/K's for kings

You must populate this grid within the nested loops, not afterwards in a secondary nested loops structure with indexing.

Sorting Algorithms

Bubble Sort

Describe how the Bubble Sort algorithm works in words. Now, write some pseudocode and a flow diagram of how you might program this sorting algorithm. Be sure to encapsulate the code inside a function!

Selection Sort

Describe how Selection Sort works in words. How many looping structures are required to run this algorithm (single, nested, etc.)? Compare this algorithm and the Bubble Sort algorithm - what is one major difference in the way it approaches the sorting problem?

Merge Sort

Describe how Merge Sort works in words. Illustrate how this works using a short example, wherein you will represent each list as a single row table. Show all the steps. Given the option between the 3 sorting algorithms listed here, which is the better/more efficient algorithm?

Recursion

What is Recursion

Write a detailed explanation of what recursion is. How does recursion function? Can you provide a few example problems that you've encountered in this course that you believe would be more easily solved with the use of recursion?

Fibonacci Sequence - Recursively

Earlier you were tasked with completing the code for the Fibonacci Sequence. Armed with that knowledge, please rewrite the code as a recursive function. You are not permitted to make use of looping control structures for this.

Associative Collections

Dictionaries VS Lists

Dictionaries are collections that differ substantially from lists. Name and describe 2 ways in which they are different. When might you employ a dictionary instead of a list?

Random Distribution Simulation

You will write a function that loops 1000 times and generates a random value between 0 and 100 each iteration. You will be using this value generated as a key to a dictionary you will initialize prior to the looping. If the key doesn't exist, then create it with a value of 1. If it does increase, then increment that value by 1 each time. Afterwards, print out the dictionary.

Classes and Class Scope

Purpose of Classes

What are classes? List 2 benefits that classes provide. When is it appropriate to use a class over a set of functions and global variables?

Student Class

Write a Student Class. The student must have a student ID, a first name, a last name, an age, and a dictionary of classes being taken. The dictionary's key should be the class code and the value should be the current grade. Your init function should take all these values as parameters. You must include getters and setters for each of the following: studentID, name (only getter), first name, last name, and age.
Other requirements:

(a) getGrade() method that returns either the grade of the class or False if it doesn't exist

(b) setGrade() method that sets the grade of class if it exists

(c) addClass() method that adds a class to list with given grade