CS 118 Sample Exam 3 Questions

Part 1 – Written

This part is to be done on paper and is closed book, closed notes, closed computer, and no phone.

You should try reviewing the rules FIRST and answering these BEFORE you seek help from either the computer or another person.

1. Fix the 5 **syntax** errors in the following program.

```
def displayMonthInfo(monthList, dayList)
   i = 0
   for month in monthList:
   print(month, dayList[i])
   i = i + 1


############## Main Logic ##################
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']
days = [31, 28, 31  30, 31, 30]
displayMonthinfo(months days)
```

2. Fix the 5 **execution** errors, NOT syntax or logical errors, in the following program.

```
def menu():
   print("Enter 1 to display class data")
   print("Enter 2 to total number of students")
   resp = input("Please make a choice: ")
   return resp


def displayClassInfo(classes, numStudentsPerClass):
   i = 0
   for className in classes:
      numStudents = numStudentsPerClass(i)
      print(className, numStudents)
      i = i + 1


def displayTotalStudents(numStudentsPerClass):
   for numStudents in numStudentsPerClass:
      total = total + numStudents
   print("Total students:", total)
############## Main Logic ##############
classList = ['CS118_01', 'CS118_02', 'CS223_01', 'CS344_01']
numStudentList = [29, 30, 25, 30]
ch = menu(classList)
if ch == '1':
   displayClassInfo(classList)
elif ch == '2':
   displayTotalStudents()
```

3. Fix the 7 **logical** errors in the attached **sampleExam3Logical.py** program.

4. Step through (trace) the following Python program showing what is in memory and what is displayed **EXACTLY** on the screen. (DON'T USE THE COMPUTER TO FIGURE THIS OUT. TRACE THROUGH IT YOURSELF.)

```
student = {'name': 'Daisy', 'credits': 102, 'gpa' : 3.9}
print("Name: ", student['name'])
print("Number of credits: ", student['credits'])
print("GPA: ", student['gpa'])
student['gpa'] = round(student['gpa'] + 0.05, 2)
print()
print("Name: ", student['name'])
print("GPA: ", student['gpa'])
print()
student['classification'] = 'Junior'
print(student)
```

5. Step through (trace) the following Python program showing what is in memory and what is displayed **EXACTLY** on the screen. (DON'T USE THE COMPUTER TO FIGURE THIS OUT. TRACE THROUGH IT YOURSELF.)
If the user is prompted for input, enter Bugs.
NOTE: **To be able to do this step through you MUST STUDY section 9.1 in your text FIRST!**

```
studentNames = ['Daisy', 'Bugs', 'Minnie']
studentGPAs = [3.9, 4.0, 3.75]
students = {}
i = 0
for name in studentNames:
    gpa = studentGPAs[i]
    students[name] = gpa
    i = i + 1

for key in students:
    print(key, students[key])

print()
nameToRemove = input("Which student would you like to remove? ")
if nameToRemove in students:
    studentGPA = students.pop(nameToRemove)
    print("You removed", nameToRemove, "with gpa", studentGPA)
else:
    print(nameToRemove, "is not in the class.")

print()
for key in students:
    print(key, students[key])
```

Part 2 – Hands-on Programming

Start up a Python IDLE for the latest Python version (3.x where x is some number).

Write a Python program, named **weightWatchersWithFunctions.py**, which is going to read a CSV file named **weightWatchers.csv** containing the following data on people at a gym: name, starting weight (lbs), current weight (lbs).

If you take a look inside the file, you will notice that it has a header row.

The structure of the program follows:

Define a function that will get the data from the CSV file. The function must have two parameters, the file name, and a name that will refer to the list of data read in from the file.

Define a function that will display a menu with the following options and return the user's choice:

1. See all the data in the list.
2. Search for a name in the list.
3. Q. Quit

Define a function that will display all of the data **in neatly formatted rows and columns below an appropriate heading**. The function should have one parameter, the list of all of the data.

Define a function that will get a name to search for from the user and return a list containing the data for all the names that match the user's input. The function should have one parameter, the list of all of the data.

Define a **main** function with the following logic:

Create an empty list and call the function that gets the data from the CSV file. Pass this function the name of the file and the empty list.

Call the menu function and collect what it returns.

While the user's choice is not Q and not q do:

Determine what the user's choice was and call the appropriate functions. Note that you can call the display function on the list that is returned by the search function.

Call the menu function again and collect what it returns

Call the main function outside of every other function.