

**Csci 1523**  
**Inclass Laboratory 3A**

**Lab Partner 1(Print):** \_\_\_\_\_  
**Lab Partner 2(Print):** \_\_\_\_\_

---

This lab contains ?? pages (including this cover page) and ?? problems. Check to see if any pages are missing. Enter all requested information on the top of this page, and put your initials on the top of every page, in case the pages become separated.

You may use your books, notes, calculator or internet sources while completing this laboratory.

Please try to answer the sections clearly and PRINT your answers legibly Please follow these guidelines when completing the sections:

This lab exercise provides practice with boolean expressions and control structures in Python. You will work with a partner on this exercise during your lab session. Two people should work at one computer. Occasionally switch the person who is typing. Talk to each other about what you are doing and why so that both of you understand each step.

## Laboratory Exercises

1. The effective use of control structures is an important component to developing efficient and maintainable programs and systems. In this section we will review some basic relational operators and their application in creating logic structures.

(a) **Boolean Expressions** Examine the Python program below and predict the values which will be displayed.

1. AAA = "A"
2. BBB = "B"
3. CCC = 1
4. DDD = 11
5. EEE = 13
6. FFF = 5
7. GGG = 4.5
8. HHH = 1.8e-40

Evaluate the following Python assignments, write the result in the space provided:

```
A = 'A' <= BBB <= 'Z' _____  
B = GGG < HHH or EEE < FFF _____  
C = not AAA >= BBB _____  
D = not AAA < BBB or HHH > 0.0 _____  
E = DDD >= GGG or GGG >= 10 _____  
G = DDD == 10 or 11 _____
```

- (b) After completing the above write a program to check your predictions. If any of your answers are incorrect, rework the appropriate questions.
2. Consider the python code listings shown below and complete the questions which follow:

(a) Code Snippets

Listing 1: *if* structure and relational operators

```
# Sample logic structures  
H = 10  
I = 5  
if H % 3 == 0  
5     H -= 1  
else:  
     H += 3  
if (H // 2 == 4) or (H % 2 == 0):  
     H += 1  
10     I = H + 10 * 3 // 5  
     I += 2
```

i. The value of H is: \_\_\_\_\_

ii. The value of I is: \_\_\_\_\_

Listing 2: *if* structure and relational operators

```
# basic control and relational operators:
J = 0
K = 25
5  if not (J == 0):
    J -= 1
    else:
        J +=1
        K //= K // 2
```

i. The value of J is: \_\_\_\_\_

ii. The value of K is: \_\_\_\_\_

Listing 3: *if* structure and relational operators

```
# basic control and relational operators
L = 20
if (L != 20) or (L % 10 == 0):
5   L += 1
    if L % 2 == 0:
        L += 5
    else:
        L += 7
10  L -= 2
    if L % 5 > 2:
        L -= 4
        L += 3
L -= 8
```

i. The value of L is: \_\_\_\_\_

- (b) After completing the above write a program to check your predictions. If and of your answers are incorrect, rework the appropriate questions.

### 3. Program with control structures

Develop a Python program which will calculate the sum of squares in an integer series, as described below.

The program will prompt the user to enter the initial integer number in the series and then the number of terms in the series. It will use repetition to compute the sum of the series, and then display the results. For example, if the initial integer 2 and the number of terms is 4:

$$Sum = 2^2 + 3^2 + 4^2 + 5^2 = 54$$

The program will use repetition to compute the sums (rather than a closed form equation). After all, the point is to practice with nested control constructs

You may assume that the user enters integer numbers when prompted for numeric values.