

# CSI Lab 03

Tuesday, January 28th

## Objectives:

- working with conditionals and selection statements
- 

## Introduction

As preparation for this week's lab you were supposed to read the first part of section 2.1 (pp81-90). First of all, if you have not done that, you should stop and look at that material.

The topic for this week and for this lab is the idea of conditionals - statements that evaluate to either True or False - and selection statements - program control structures that decide whether or not to perform some action based on the result of one or more conditionals. While there are several selection statements available to us in Python, this lab will focus on two of them:

- single-way selection statement (if)
  - two-way selection statement (if/else)
- 

## Activity A

Table 2.1 in your textbook (page 83) presents the six base Boolean Operators that we use as a starting point for writing conditionals. For the most part these are operators that you have used in your math classes although the format may be a little different than you are used to seeing.

[Q1] **Predict** what will happen when you invoke the following expressions. **After** making your prediction, enter the statement at the interactions prompt and check if you were correct. Warning, one of these statements is invalid and will produce an error.

Expression	Predicted Result	Actual Result
------------	------------------	---------------

10 < 11		
10 > 11		
12 < 12		
12 <= 12		
12 >= 12		
12 = 12		
12 == 12		
13 == 12		
13 != 12		

[Q2] **Predict** what will happen when you invoke the following statements. **After** making your prediction, enter the statement at the interactions prompt and check if you were correct.

Command	Predicted Result	Actual Result
print ('a' < 'b')		
print ('A' < 'B')		
print('A' < 'b')		
print( 'a' < 'B')		

[Q3] Enter each of the following commands in to the command prompt in the interactions window. Record the results in the table below.

Command	Actual Result
print (ord('A'))	
print (ord('B'))	
print (ord('a'))	
print (ord('b'))	

[Q4] Using what you observed in your responses to Q3, what do you **THINK** is happening when you make comparison operators such as in Q2?

---

## Activity B

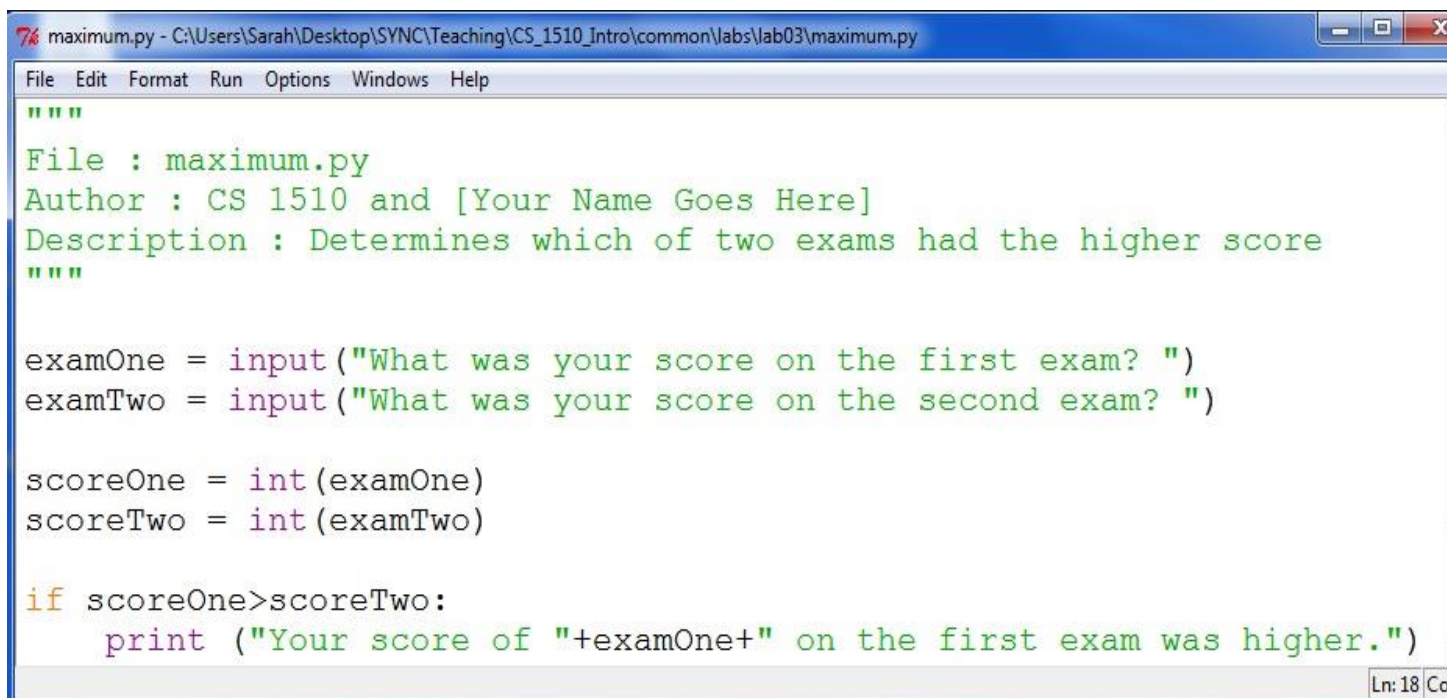
The logic operations/statements that you performed in Activity A are known as "conditional statements." They are called this because they check to see if some condition is True or False. We often use conditional statements inside of a slightly larger statement called a selection statement. A selection statement is a statement that decides whether or not a particular action can be taken.

You are contacted by Dr. Chris P. Bacon, a professor at Whatsammata University (aka, Whatsammata U). He seems to have very little common sense and needs help with the easiest of tasks. He comes to you with a request. He would like a program that asks each student what they received on the first two exams in his class and print out a message about which score was higher.

You think about this for a little bit and decide that you can do that by writing a script (called "maximum.py") that

- prompts the user for their score on the first exam
- prompts the user for their score on the second exam
- checks to see if:
  - the score on the first exam was larger than the score on the second exam
  - the score on the second exam was larger than the score on the first exam
  - the scores were identical
- prints an appropriate message either way

After thinking about this you write the following code:



```
7% maximum.py - C:\Users\Sarah\Desktop\SYNC\Teaching\CS_1510_Intro\common\labs\lab03\maximum.py
File Edit Format Run Options Windows Help
"""
File : maximum.py
Author : CS 1510 and [Your Name Goes Here]
Description : Determines which of two exams had the higher score
"""

examOne = input("What was your score on the first exam? ")
examTwo = input("What was your score on the second exam? ")

scoreOne = int(examOne)
scoreTwo = int(examTwo)

if scoreOne>scoreTwo:
    print ("Your score of "+examOne+" on the first exam was higher.")
```

When you run this you want to see something like:

```
What was your score on the first exam? 85
What was your score on the second exam? 79
Your score of 85 on the first exam was higher.
>>> ===== RESTART ===
>>>
What was your score on the first exam? 84
What was your score on the second exam? 88
Your score of 88 on the second exam was higher.
>>> ===== RESTART ===
>>>
What was your score on the first exam? 92
What was your score on the second exam? 92
Your two exam scores were exactly the same.
>>>
```

Copy this code in to a file called `maximum.py` and run what you have so far. You will notice that the first case works, but that the second two situations don't work yet. Add the code necessary to complete this program.

[SIG1] When you have this code working properly please demonstrate it for one of the instructors.

---

## Activity C

Dr. Chris P. Bacon calls you again. He needs help advising his CS1 students. Recall that you need to get a C in this class to pass the course and move on to data structures. By the definition in my syllabus, a C will have a cutoff no HIGHER than 72% (that is, a score of 72 or better is guaranteed to be a C in this course).

Write a script (called "pass.py") that

- prompts the user for the total number of points possible in the class
- prompts the user for the number of points they have earned
- calculates their percentage
- checks to see if that percentage is "passing" or not
- prints an appropriate message either way

For example:

```

How many total points were possible? 850
How many points did you earn? 791
That is 93.05882352941175%
You can enroll in data structures
>>> ===== RESTART
>>>
How many total points were possible? 850
How many points did you earn? 597
That is 70.23529411764706%
You should retake CS1

```

Use Code Listing 2.6 (on page 89 of your textbook) and the code from Activity B as a guideline to help you with this activity.

[SIG2] When you have this code working properly please demonstrate it for one of the instructors.

## Activity D

In Activity A you were asked to consider the six base Boolean operators. In addition to those six there are three more that we often use in computer programming. Although I have not formally assigned the reading yet, p 109 in your textbook has a discussion about these operators.

[Q5] **Predict** what will happen when you invoke the following expressions. **After** making your prediction, enter the statement at the interactions prompt and check if you were correct.

Expression	Predicted Result	Actual Result
10 < 11 and 11 < 12		
10 < 11 and 13 < 12		
13 < 12 and 10 < 11		
13 < 12 and 11 < 10		
10 < 11 or 11 < 12		
10 < 11 or 13 < 12		
13 < 12 or 10 < 11		
13 < 12 or 11 < 10		
not 10 < 11		

not (10 < 11)		
not 13 < 12		
not (13 < 12)		

---

## Activity E

Dr. Chris P. Bacon knows that to be admitted to Whatsamatta U. you must have a score of at least 23 on the ACT as well as a high school GPA of at least 1.75.

Write a script (called "admit.py") that

- prompts the user for their score on the ACT exam
- prompts the user for their high school GPA
- checks to see if the student should be admitted or not
- prints an appropriate message either way

For example:

```
What was your score on the ACT? 25
What was your high school GPA? 2.35
Congratulations. You can attend Whatsamatta U.
>>> ===== RESTART ===
>>>
What was your score on the ACT? 23
What was your high school GPA? 2.35
Congratulations. You can attend Whatsamatta U.
>>> ===== RESTART ===
>>>
What was your score on the ACT? 22
What was your high school GPA? 2.35
Sorry. You don't live up to our standards.
```

[SIG3] When you have this code working properly please demonstrate it for one of the instructors.

---

## Finishing Up

## **Congratulations!**

When you are ready to leave, make sure that your activity log is completed before submitting it to Dr. Diesburg.

**Do not forget to “log off” of the machine you are working on and TAKE YOUR FLASH DRIVE WITH YOU.**