ICS3C0 Module B.1 Python Programming Introduction Name:

**Level 3: Lists & Logic**

Access the Tutorial and start at “Lesson 7: Booleans”

Questions

1. Complete “Lesson 7: Booleans – AND Comparisons” by typing the sample commands in the black area of the IDE.

a. Try the following Python statements and record the results.

i. True and True

**=> True**

ii. True and False

**=>False**

iii. False and True

**=>False**

iv. False and False

**=>False**

b. Explain if there are any other combinations of True / False.

**No, there are no other combinations of True / False. All the options have been exhausted.**

c. Explain how the AND operator is similar to a math operator and how it is different.

**The “and” operator takes into account both expressions. Both sides of the expression need to be true; in that way, they are similar. However, the way in doing so is different seeing as you’ll need the answer(s) beforehand to complete the expression(s).**

2. Complete “Lesson 7: Booleans – OR Comparisons” by typing the sample commands in the black area of the IDE.

a. Try the following Python statements and record the results.

i. True or True

**=>True**

ii. True or False

**=>True**

iii. False or True

**=>True**

iv. False or False

**=>**

b. Explain how the OR operator is similar to the AND operator and how it is different.

**Like the “and” operator before it, it takes into account both options. The difference between the two is “or expressions” are true if either portions of the expression are true, compared to the “and” operator’s false answer if one of them is true.**

3. Complete “Lesson 7: Booleans – NOT Comparisons” by typing the sample commands in the black area of the IDE.

a. Try the following Python statements and record the results.

i. not (True or True)

**=>True**

ii. not (True or False)

**=>False**

iii. not (False or True)

**=>False**

iv. not (False or False)

**=>True**

b. Explain how the combination of the NOT & OR operators is similar to the AND operator by itself and how it is different.

**If both sides of the expression are true, the answer is true, which the “not” operator can turn into the opposite operator. Depending on how you word the question given to Python, the “and” operator can do something similar. Though, there can also be a difference between the two whether the answer is the opposite or not.**

4. Complete “Lesson 7: Booleans – Expressions” by typing the sample commands in the black area of the IDE.

a. Explain why the following two Python statements give different results.

i. not (True or True)

ii. not True or True

**Parentheses gives way for the bracketed characters to be prioritized. As such, “True or True” is true, but because the opposite of “not true” is false and not is there along with the bracketed expression, the answer is false.**

b. Explain why the following two Python statements give the same results.

i. not (True and True)

ii. not True and True

**Regardless of whether or not the parentheses is there, the answer is false because of this: the “and” operator. To return a “True” statement, both sides of the expression need to be true. Bracketed characters allow for priority and “True and True” is true. However, because the “not” operator is in the expression outside the parentheses and the “and” operator only returns a true statement if both sides are true, after solving the parentheses the opposite of “not true” is false, returning a “false” answer.**

5. Complete “Lesson 7: Booleans – Practice” by typing the sample commands in the black area of the IDE.

a. Create three more practice expressions similar to those in the tutorial.

**False or True and True, “Mario”== “Mario” and True, 1==7**

b. Provide the results for your practice expressions

**True, True, False**

6. Complete “Lesson 8: Lists – A Collection of Objects” by typing the sample commands in the black area of the IDE.

a. Create a list of your favorite sports teams.

**TOR (Toronto Raptors), HOU (Houston Rockets)**

b. Assign your list to a variable.

**Basketball**

c. Confirm that your variable and your list are the same.

**Yes, they are the same.**

7. Complete “Lesson 8: Lists – List Indexes” by typing the sample commands in the black area of the IDE.

a. What is the list index of the last team in your list of favorite sports teams.

**0 = TOR, 1 = HOU.**

b. In the tutorial, the error produced by typing “fruit[3]” is an example of:

i. A Syntax Error?

ii. A Runtime Error?

iii. A Logic Error?

**An IndexError.**

8. Complete “Lesson 8: Lists – Practice” and “Lesson 8: Lists – Practice Answers” by typing the sample commands in the black area of the IDE.

NOTE: Starting with Lesson 9 you should use the WHITE area of the IDE for entering example code with multiple statements.

9. Complete “Lesson 9: Logic – Making Decisions” by typing the sample commands in the white area of the IDE.

a. Modify the tutorial code to print “Hi Alfred!” based on a decision using numbers

**myname = "Alfred"\*1**

**if myname == "Alfred"\*1:**

**print("Hi, Alfred"\*250)**

10. Complete “Lesson 9: Logic – Adding A Choice” by typing the sample commands in the white area of the IDE.

a. Modify the tutorial code to print your first name or your last name based on a choice (using “else”).

**myname = "James"**

**if myname == "James":**

**print("Hey, James!")**

**else:**

**print ("Wait a minute... You're not James!")**

11. Complete “Lesson 9: Logic – Adding Many Choices” and “Lesson 9: Logic – Practice” by typing the sample commands in the white area of the IDE.

a. Modify the tutorial code and “elif” statements to make a choice using at least 4 of your friends names.

**myname = "James"**

**if myname == "James":**

**print("Hey James!")**

**if myname == "James":**

**print("Hey, Amadaeus, Carlos, Angelie and Tiana!")**