ICS3C0 Module B.1 Python Programming Introduction Name:

**Outline**

Access the Python Development environment and follow the tutorial to gain an initial exposure to a programming language. Begin to develop an familiarity with basic programming concepts.

**Objectives**

· Use correct terminology to describe programming concepts;

· Describe the types of data that computers can process and store (e.g., numbers, text);

· Explain the difference between constants and variables used in programming;

· Use variables, expressions, and assignment statements to store and manipulate numbers and text in a program

**Materials**

· Python3 Development Environment at: //repl.it/

· Python Tutorial at:<http://www.letslearnpython.com/learn/>

**Accessing the Python3 Web IDE Environment**

Accessing the IDE

· Go to:<https://repl.it/>

· Select Python3

· Sign-up / Create an account

· Make sure you can remember your account information for the rest of the course.

Using the IDE

· Use the black area like a calculator to try simple statements or commands

· Use the white area to create programs with multiple statements

**Accessing the Tutorial**

Accessing the Tutorial

· Go to:<http://www.letslearnpython.com/learn/>

· Read up to “Lesson 3: Math”

**Level 1: Basic Math & Strings**

Access the Tutorial and start at “Lesson 3: Math”.

Questions

1. Complete “Lesson 3: Math – Math Basics” by typing the sample commands in the black area of the IDE.

a. Create your own expression using 5 “+” and “-“ operators.

b. List your expression and the result below.

**5+9-3=11**

2. Complete “Lesson 3: Math – More Operators” by typing the sample commands in the black area of the IDE.

a. Create your own expression using 5 “\*” and “/” operators.

b. List your expression and the result below.

**5\*9/3=15.0**

3. Complete “Lesson 3: Math – More Division” by typing the sample commands in the black area of the IDE.

a. Create one division expression that gives a whole number answer

**20/4**

b. And one division expression that gives a decimal number answer.

**16/14**

c. List your expressions and the results below.

**20/4=5**

**16/14=1.1428571428571428**

4. Complete “Lesson 3: Math – Floats” by typing the sample commands in the black area of the IDE.

a. Use the “round()” function for the expressions you created in question #3 above.

b. List your “round()” expressions and the results they return below.

**round(20/4)=5**

**round(16/14)=1.1428571428571428**

5. Read through “Lesson 3: Math – Comparison Operators”.

a. Why do you think Equals is “==” instead of “=”?

**Because Python doesn’t understand what “=” means. It’s an invalid command.**

b. What does “=” mean?

**It’s an assignment operator. It assigns a value to a variable**

6. Complete “Lesson 3: Math – Practice” and “Lesson 3: Math – Practice Answers” by typing the sample commands in the black area of the IDE.

a. Create an expression using 5 different operators that returns a “True” result

**7==7!=9>3<=3>=3**

b. And an expression using 5 different operators that returns a “False” result.

c. List your expressions and the results returned below.

**8==4<=17>23!=23>=71**

7. Complete “Lesson 4: Strings – Strings” and “Lesson 4: Strings – Examples” by typing the sample commands in the black area of the IDE.

a. Explain why typing “apple” works and why typing apple without quotes gives an error.

**If you want Python to read a string, it must be inside quotes.**

b. Also explain why “2 + 5” does not equal 7.

**When putting something in quotation marks, Python is reading a string—not adding.**

8. Complete “Lesson 4: Strings – Operators” by typing the sample commands in the black area of the IDE.

a. Explain why typing “appl” + “e” works and why typing “apple” - “e” gives an error.

**You can’t simply subtract string\s with the “-” operation.**

b. Also explain why “Hello” \* 10 works but why “Hello” / 10 does work.

**You can’t divide a string. Python is simply reading “Hello” and it’s a word. You can’t divide words.**

9. Complete “Lesson 4: Strings – Indexes” by typing the sample commands in the black area of the IDE.

a. List the letters in your first name and the index for each letter in your first name.

James. J=0 a=1 m=2 e=3 s=4

10. Complete “Lesson 4: Strings – Indexes Examples” by typing the sample commands in the black area of the IDE.

a. Explain why print(“Hello!”[4]) does not print “l”.

**Because the index starts from 0 and counting to four from 0 prints “o”**

b. What does print(“Hay, Bob!”[4]) print?

**The space between “Hey” and “Bob”.**

For a hint try print(“Hay, Bob!”[3]) and print(“Hay, Bob!”[5])

11. Complete “Lesson 4: Strings – Rules” by typing the sample commands in the black area of the IDE.

a. Explain why print(“Hello!”[7]) gives an error.

**The string index is out of range. In other words, “Hello” is 5 indexes and seven indexes is past the limit. You’re asking for something that isn’t there.**

**Level 2: Booleans & Variables**

Access the Tutorial and start at “Lesson 5: Variables”

Questions

1. Complete “Lesson 5: Variables – Save a Value” by typing the sample commands in the black area of the IDE.

a. What do you get if you type puppies / 3?

**12.0.**

b. Why doesn’t typing kittens / 3 work?

**Because we didn’t assign kittens a value, let alone the number 36.**

2. Complete “Lesson 5: Variables – Assign a New Value” by typing the sample commands in the black area of the IDE.

a. Explain how the following sequence of commands works:

· puppies = 36

· puppies = puppies / 6

· puppies

**Because you correctly assign the variable “puppies” to a number: 36. Python stores this, unless you change the value given to “puppies”. Every time you use the variable “puppies” it’s give you “36”. Dividing puppies equals 6.0 due to the numerical value (36) given to “puppies”.**

3. Read through “Lesson 5: Variables – Rules”.

4. Complete “Lesson 5: Variables – Math Operators” by typing the sample commands in the black area of the IDE.

a. Explain what happens for following sequence of commands:

· colour = “red”

· puppies = 36

· colour + puppies

An error occurs. “Puppies” must be a string, not an integer.

5. Complete “Lesson 5: Variables – String Operators” by typing the sample commands in the black area of the IDE.

a. Explain why the following commands give different results:

· Color + day \* fishes

· ( Color + day ) \* fishes

**The expression “Colour + day \* fishes” equals “yellowMondayMondayMonday” because Python orders things according to order of operations. Multiplication takes precedence over addition by default when encountering this problem in Python. Putting parentheses around “colour” and “day” ensures that both those string values take precedence over the multiplication, providing a different answer.**

6. Complete “Lesson 5: Variables – Indexes” by typing the sample commands in the black area of the IDE.

a. What is the index of ‘r’ in “watermelon”?

**4.**

b. Write an expression using mynumber to return ‘r’

**My number = 5**

**Fruit [mynumber - 1]**

**=> “r”**

7. Complete “Lesson 5: Variables – Assignments or Comparisons” by typing the sample commands in the black area of the IDE.

a. What is the difference between “=” and “==”?

**When we're assigning a value, we're saying "this equals that". That's a short sentence, so it only gets one equal sign: =**

**But when we're comparing values, we're asking "is this thing equal to that thing?". And that's a longer sentence, so it gets two equal signs: ==.**

b. Create your own mnemonic to remember this difference.

**Double the equal signs, double the length (==).**

8. Complete “Lesson 6: Errors – Examples” by typing the sample commands in the black area of the IDE.

a. What doesn’t “friend” + 5 work?

Because I can only add strings to other strings—not other numbers (unless that number is given a value itself).

b. What is the difference between int and str?

“Int” stands for “integer” while “str” stands for “string”. Integers are, in Python’s case, regular whole numbers. Python cannot concatenate objects of different types. We tried to concatenate two pieces of data (which Python calls objects) - the string "friend" and the number 5. But Python isn’t able to concatenate those two objects because one is a string and one is an integer. They're two different types of data. And so we get what’s called a "TypeError".  
  
When Python sees an expression like this, it doesn't know if we're trying to add numbers or concatenate strings. And Python is not smart enough to guess what we mean - we have to tell it. So with this error, Python is letting us know that it needs clarification.

9. Read through “Lesson 6: Errors – Parts of an Error Message”.

a. Is “friend” + 5 an example of:

i. A Syntax Error?

ii. A Runtime Error?

iii. A Logic Error?

**A TypeError.**

10. Read through “Lesson 6: Errors – Fixing Errors”.

a. Use the ‘print’ command to print your first name and last name.

**James Yeboah**

11. Complete “Lesson 7: Booleans – Types of Data” by typing the sample commands in the black area of the IDE.

a. What is the value of: type(“True”)?

**A string.**

b. What is the value of: type( True )

**A boolean.**

c. Why is the result different?

**A boolean only has two possible values - it can be either True or False.**

12. Complete “Lesson 7: Booleans – What Is A Boolean” by typing the sample commands in the black area of the IDE.

a. Why do you think that having a Boolean data type is important in computer programming?

**Booleans are used in programming a lot when you need to make decisions about what to do in your code:**

**"If this expression is True, do something; if the expression is False, do something else instead."**

13. Complete “Lesson 7: Booleans – Trying Out Booleans” by typing the sample commands in the black area of the IDE.

a. Why do you think that there is no “Maybe” Boolean data value in computer programming?

**A boolean only has two possible values: “True” and “False”. Nothing less, nothing more.**