Assignment 1

January 21, 2020

1 Assignment 1 CSC 101

1.1 Problem 1

In the code cell below, enter a python command which will print out the string "My name is ...". Replace the three dots with your own name.

[1]:
```
print("My name is Paul.")
```

```
My name is Paul.
```

1.2 Problem 2

The general rule in evaluating mathematical operations is that when operators are of the same order of precedence, evaluation proceeds from left to right. I want you to examine whether this is true for the case of exponentiation. Use the integers 2, 3, and 5 in the framework in the following paragraph.

In the code cell below, enter three assignment statements. The first one should create the variable no_parens. The second one should create the variable left_parens. The third one should create the variable right_parens. Then use three print() statements to show the values of these three variables.

Describe your conclusion in the markdown cell below the code cell.

[45]:
```
# Example one shows order of operations and parentheses// this was a wrong␣
 ↪attempt : See example two for correct exponent example...
x = 2**1 + 3**1 * 5**1
no_parens = str(x)

y = (2**1 + 3**1) * 5**1
left_parens = str(y)

z = 2**1 + (3**1 * 5**1)
right_parens = str(z)

# no_parens, left_parens, right_parens
print("Example one: ",no_parens,",",left_parens,",",right_parens)
```

```
#Example two shows exponents prioritizing from right to left: this is the␣
 ↪correct example
a = 2**3**5
no_parens2 = str(a)

b = (2**3)**5
left_parens2 = str(b)

c = 2**(3**5)
right_parens2 = str(c)

# no_parens2, left_parens2, right_parens2
print("Example two: ",no_parens2,",",left_parens2,",",right_parens2)
```

```
Example one:  17 , 25 , 17
Example two:
14134776518227074636666380005943348126619871175004951664972849610340958208 ,
32768 ,
14134776518227074636666380005943348126619871175004951664972849610340958208
```

The accepted practice for order of precedence is the "PEMDAS" method: Parentheses, Exponents, Multiplication, Division, Addition, Subtraction. Values that are on the same level i.e. "M/D" and "A/S" are done from left to right. This above example shows that Exponentiation is completed from right to left. The no_parens2 and the right_parens2 examples match in value and the left_parens2 computed differently.

1.3 Problem 3

Give the variable decimal the value 1.0 and the variable whole the value the value 2. Then put the sum of these two variables in the variable the_sum. Print the value of the_sum. Print the type of the_sum returned by the type() function. Do your work in the following code cell.

[34]:
```
decimal = 1.0
whole = 2
the_sum = decimal + whole
print(the_sum,type(the_sum))
```

```
3.0 <class 'float'>
```

1.4 Problem 4

Give the variable first_name your first name and the variable last_name your last name. Create the variable normal_name by concatenating your names separated by a single blank space. Create the variable reversed_name by putting your last name first and your first name last. Separate your two names with a comma and a blank space. Print both versions. Do your work in the following code cell.

[53]:
```
first_name = "Paul"
last_name = "Schultz"
reversed_name = last_name + ", " + first_name
normal_name = first_name + " " + last_name
print("My Normal name is", normal_name)
print("My Reversed name is", reserved_name)
```

```
My Normal name is Paul Schultz
My Reversed name is Schultz, Paul
```

[54]:
```
first_name = "Paul"
last_name = "Schultz"
reversed_name = last_name + ", " + first_name
normal_name = first_name + " " + last_name
print("My Normal name is", normal_name)
print("My Reversed name is", reserved_name)
```

```
My Normal name is Paul Schultz
My Reversed name is Schultz, Paul
```

[0]:
```

```