Practice Problem Set – Data Types, Variables & Expressions

1. a. Write python program to produce the following pattern at the output:

```
        1
      2 2
    3 3 3
  4 4 4 4
5 5 5 5 5
```

   b. Save the file with *.txt* extension instead of *.py* extension. What difference do you see in code and execution?

   c. Write a Python program containing exactly one print statement that produces the above output

2. Will the following lines of code print the same thing? Explain why or why not.

```
x = 6
print(6)
print("6")
```

3. What happens if you attempt to use a variable within a program, and that variable has not been assigned a value?

4. What is the difference between the following two strings? `'n'` and `'\n'`?

5. What is wrong with the following statement that attempts to assign the value ten to variable `x`?

```
10=x
```

6. Given the following assignment:

```
x = 2
```

   Indicate what each of the following Python statements would print.

   a. `print("x")`

   b. `print('x')`

   c. `print(x)`

   d. `print("x + 1")`

   e. `print('x' + 1)`

   f. `print(x + 1)`

7. Given the following assignments:

```
i1 = 2     i2 = 5     i3 = -3     d1 = 2.0     d2 = 5.0     d3 = -0.5
```

   Evaluate the result and type of result for each of the following Python expressions.

   a. `i1 + (i2 * i3)`

   b. `i1 * (i2 + i3)`

   c. `i1 / (i2 + i3)`

   d. `i1 // (i2 + i3)`

   e. `i1 / i2 + i3`

   f. `i1 // i2 + i3`

   g. `3 + 4 + 5 / 3`

   h. `3 + 4 + 5 // 3`

   i. `(3 + 4 + 5) / 3`

   j. `(3 + 4 + 5) // 3`

   k. `d1 + (d2 * d3)`

   l. `d1 + d2 * d3`

   m. `d1 / d2 - d3`

   n. `d1 / (d2 - d3)`

   o. `d1 + d2 + d3 / 3`

   p. `d1 + d2 + d3) / 3`

   q. `d1 + d2 + (d3 / 3)`

   r. `3 * (d1 + d2) * (d1 - d3)`

8. What is an integer equivalent to `True` in Python?

9. What is the integer equivalent to `False` in Python?

10. Is the value `-16` interpreted as `True` or `False`?

11. Given the following definitions:

```
x, y, z = 3, 5, 7
```

Evaluate the following Boolean expressions:

a. `x == 3`
b. `x < y`
c. `x >= y`
d. `x <= y`
e. `x != y - 2`
f. `x < 10`
g. `x >= 0 and x < 10`
h. `x < 0 and x < 10`
i. `x >= 0 and x < 2`
j. `x < 0 or x < 10`
k. `x > 0 or x < 10`
l. `x < 0 or x > 10`

12. Given the following definitions:

```
x, y = 3, 5
b1, b2, b3, b4 = True, False, x == 3, y < 3
```

Evaluate the following Boolean expressions:

a. `b3`
b. `b4`
c. `not b1`
d. `not b2`
e. `not b3`
f. `not b4`
g. `b1 and b2`
h. `b1 or b2`
i. `b1 and b3`
j. `b1 or b3`
k. `b1 and b4`
l. `b1 or b4`
m. `b2 and b3`
n. `b2 or b3`
o. `b1 and b2 or b3`
p. `b1 or b2 and b3`
q. `b1 and b2 and b3`
r. `b1 or b2 or b3`
s. `not b1 and b2 and b3`
t. `not b1 or b2 or b3`
u. `not (b1 and b2 and b3)`
v. `not (b1 or b2 or b3)`
w. `not b1 and not b2 and not b3`
x. `not b1 or not b2 or not b3`
y. `not (not b1 and not b2 and not b3)`
z. `not (not b1 or not b2 or not b3)`

13. Express the following Boolean expressions in simpler form; that is, use fewer operators or fewer symbols. `x` is an integer.

a. `not (x == 2)`
b. `x < 2 or x == 2`
c. `not (x < y)`
d. `not (x <= y)`
e. `x < 10 and x > 20`
f. `x > 10 or x < 20`
g. `x != 0`
h. `x == 0`

14. Express the following Boolean expressions in an equivalent form without the `not` operator. `x` and `y` are integers.

a. `not (x == y)`
b. `not (x > y)`
c. `not (x < y)`
d. `not (x >= y)`
e. `not (x <= y)`
f. `not (x != y)`
g. `not (x != y)`
h. `not (x == y and x < 2)`
i. `not (x == y or x < 2)`
j. `not (not (x == y))`

15. Attempt exercises 2.11 through 2.17 from text book1: *Introduction to Computing Using Python, 2e, Ljubomir Perkovic*.