Lab Exercise 2. Variables, assignment, types and operators

1. What is a Python comment? How do you indicate a comment? What purpose does it serve?

A Python comment is a piece of text in your code that is not executed as part of the program but is there for documentation or clarification purposes.

To indicate a comment in Python, you can use the hash symbol #. Anything following the # on the same line is treated as a comment.

The purpose of comments is to provide explanations, notes, or context to your code, making it more understandable for yourself and others who read it.

2. What is a namespace in Python?

In Python, a namespace is a container that holds a mapping of names (identifiers) to objects. It is a fundamental concept in Python's naming system and plays a crucial role in organising and managing variables, functions, classes, and other objects in a Python program. Namespaces help prevent naming conflicts and provide a way to uniquely identify and access different elements in your code.

3. Whitespace:

(a) When does whitespace matter?
(b) When does whitespace not matter?

Whitespace, such as spaces, tabs, and line breaks, has specific significance in Python.

(a) Indentation: Whitespace at the beginning of lines (indentation) matters significantly in Python. It is used to define the structure and scope of code blocks like loops, conditionals, functions, and classes. Incorrect indentation can lead to syntax errors or logical errors.

(b) Whitespace in Expressions: Whitespace between operators and operands generally doesn't matter. Python is often flexible in terms of spacing within expressions. You can use spaces or not, depending on your coding style and readability preferences.

4. Mixed operations:

(a) What type results when you divide an integer by a float? A float by an integer?
(b) Explain why that resulting type makes sense (as opposed to some other type).

(a) When you perform operations that involve mixed types in Python, the result is determined based on the Python type coercion rules. Specifically:

When you divide an integer by a float, the result is always a float.

Similarly, when you divide a float by an integer, the result is also a float.

(b) The resulting type being a float makes sense because floating-point numbers can represent a wider range of values and have more precision than integers. When performing operations involving mixed types, Python promotes the result to the type that can represent the result most accurately without loss of information.

5. Consider integer values of `a`, `b`, and `c` and the expression `(a + b) * c`. In mathematics, we can substitute square brackets, `[ ]`, or curly braces, `{ }`, for parentheses, `( )`. Is that same substitution valid in Python? Try it.

In Python, you cannot substitute square brackets [] or curly braces {} for parentheses () in the same way you might do in mathematics. Parentheses have a specific meaning in Python and are used for defining function calls and specifying the order of operations, while square brackets and curly braces have different purposes.

6. Which of the following are acceptable variable names for Python?
(a) xyzzy Acceptable.
(b) 2ndVar Not acceptable.
(c) rich&bill Not acceptable.
(d) long name Not acceptable.
(e) good2go Acceptable.

7. Give the values printed by the following program for each of the labeled lines.

```
int_a = 27
int_b = 5
int_a = 6
print(int_a) # Line 1
print(int_b + 5) # Line 2
print(int_b) # Line 3
```

(a) What is printed by Line 1? 6.
(b) What is printed by Line 2? 10.
(c) What is printed by Line 3? 5.

8. Give the values printed by the following program for each of the labeled lines, and answer the associated questions.

```
a_float = 2.5
a_int = 7
b_int = 6
print(a_int / b_int) # Line 1
print(a_int // a_float) # Line 2
print(a_int % b_int) # Line 3
print(int(a_float)) # Line 4
print(float(a_int)) # Line 5
```

(a) Line 1: What is printed? What is its type?
(b) Line 2: What is printed? What is its type?
(c) Line 3: What is printed? What is its type?
(d) Line 4: What is printed? What is its type?
(e) Line 5: What is printed? What is its type?

(a) Line 1:

What is printed? 1.1666666666666667
What is its type? float
(b) Line 2:

What is printed? 2.0
What is its type? float
(c) Line 3:

What is printed? 1
What is its type? int
(d) Line 4:

What is printed? 2
What is its type? int
(e) Line 5:

What is printed? 7.0
What is its type? float

9. Give the values printed by the following program for each of the labelled lines.

```
a_int = 10
b_int = 3
c_int = 2
print(a_int + b_int * c_int) # Line 1
print( (a_int + b_int) * c_int ) # Line 2
print(b_int ** c_int) # Line 3
```

(a) What is printed by Line 1? 16
(b) What is printed by Line 2? 26
(c) What is printed by Line 3? 9

10. Change the program below to calculate and print the area of a rectangle instead.

```
from math import pi
r = 12
area = pi * r ** 2
print("The area of a circle with radius", r, "is", area)
```

New code:

```
length = 12   # Length of the rectangle
width = 6     # Width of the rectangle
area = length * width   # Calculate the area of the rectangle

print("The area of a rectangle with length", length, "and width", width, "is", area)
```

11. Write a Python program that prompts for a number. Take that number, add 2, multiply by 3, subtract 6, and divide by 3. You should get the number you started with. Done.

12. Assignment:

```
my_int = 5
my_int = my_int + 3
print(my_int)
```

(a) If you execute the three lines of code, what will be printed? Explain your answer using the rules of assignment. 8

my_int = 5: This line assigns the value 5 to the variable my_int.my_int = my_int + 3: This line updates the value of my_int. It takes the current value of my_int, which is 5, adds 3 to it, and assigns the result (8) back to my_int. This is done through the process of assignment where the right side of the = is evaluated first, and then the result is stored in the variable on the left side. So, my_int is updated to 8.

print(my_int): This line prints the current value of my_int, which is now 8.
(b) Rewrite `my_int = my_int + 3` using the `+=` symbol.

my_int+=3

13. Assignment:

```
my_var1 = 7.0
my_var2 = 5
print(my_var1 % my_var2)
```

If you execute these three lines of code, what will be printed? 2.0

14. Prompt for input and then print the input as a string, an integer, and a float-point value. What values can you input and print without errors being generated?

For a string: Enter "Hello, World!" or any text.

For an integer: Enter "42" or any valid integer like "123".

For a float: Enter "3.14" or any valid floating-point number like "2.71828".

15. Consider the expression `(a + b) * c` , but with string values for `a`, `b`, and `c`. Enter that into the Python shell. What happens? Why?

(a + b) would concatenate the strings "Hello" and "World" to form "HelloWorld".

Then, Python attempts to multiply the concatenated string "HelloWorld" by the string "3".

However, multiplying a string by another string is not a valid operation in Python, which is why you get a TypeError. Python doesn't have a defined meaning for multiplying two strings together, so it raises an error.

16. (Integer operators) One way to determine whether an integer is even is to divide the number by 2 and check the remainder. Write a three-line program that prompts for a number, converts the input to an integer, and prints a 0 when the number is even and a 1 when the number is odd.

17. Body mass index (BMI) is a number calculated from a person's weight and height. According to the Centers for Disease Control and Prevention, the BMI is a fairly reliable indicator of body fatness for most people. BMI does not measure body fat directly, but research has shown that BMI correlates to direct measures of body fat, such as underwater weighing and dual-energy X-ray absorptiometry. The formula for BMI is

$$\text{weight} / \text{height}^2$$

where weight is in kilograms and height in meters.

(a) Write a program that prompts for metric weight and height and outputs the BMI.
(b) Write a program that prompts for weight in pounds and height in inches, converts the values to metric, and then calculates the BMI.