Answer the following 10 questions (and subparts) using information you find in Chapter 1 of “The Practice of Computing Using Python.” You can use the book’s exact text for your answers. Answers should be short, clear, and reasonably complete. The answers must be in this document.

1. According to the textbook, what is a computer program?

Answer: A program is a set of instructions that are executed sequentially by a computer, one after the other in the order in which they were typed.

1. Python is an **interpreted** language. What does “interpreted” mean in this context?

Answer: by interpreted we mean that there is a program within Python called the interpreter that takes each line of Python code, one line at a time, and executes that code. This feature allows us to try out lines of code one at a time by typing into the Python shell.

1. What is a Python *comment?* In what ways can you indicate a comment? What are some purposes for writing comments?

Answer: Comments in Python is the inclusion of short descriptions along with the code to increase its readability. Comments are indicated by typing a pound(hash) at the beginning of the line. Also it can be possible to make a multi-line comments by using docstrings (‘“ ’’’). Its purpose is to clarify the code and explains the basic logic behind why a particular line of code was written

1. In Python, what is a *namespace*?

Answer: We can think of a namespace as a dictionary in which the keys are the object names and the values are the objects themselves, so a namespace is a relation between names and objects.

1. Regarding Whitespace in python:
   1. What is whitespace? Include examples.

Answer: Whitespace is a character that is used for spacing and has an “empty” representation. Python count as whitespace the following characters: space, tab, return, linefeed, formfeed and vertical tab

* 1. When does whitespace matter?

Answer: Indentation, which is the whitespace at the beginning of a line.

* 1. When does whitespace not matter?

Answer: whitespace is ignored within both expressions and statements. In Addition, blank lines are also ignored.

1. Regarding Python statement and expressions:
   1. Explain the difference between a statement and an expression

Answer:

An expression is a combination of operators and operands that is interpreted to produce some other value (return value).

A statement is an instruction that the Python interpreter can execute, it doesn’t have a return value but it might have a side effect.

* 1. Give an example of both a statement and an expression Answer:

statement: my\_variable = 5,

expression: my\_variable = my\_variable + 7

* 1. Explain what is meant by a statement having a *side effect*

Answer: An expression is said to have a side effect if it modifies some state variable value(s).

1. Mixed operations:
   1. What data type results when you divide an integer by a float?

Answer: A float

* 1. What data type results when you divide a float by an integer?

Answer: A float

* 1. Why do the resulting data type(s) answered above make sense - as opposed to other data type(s)?

Answer: It make sense because a float data type was used in both operations, unless the result is cast into an integer, the result always will be a float.

1. 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? Explain why or why not and be specific.

Answer: No, we can’t substitute the parentheses because they have a purpose in phyton, in the given example the expression inside the parentheses is executed first then the multiplication. If we modify that line of code, then we are going to get a different output.

1. Evaluate the following 3-line Python program:

1> my\_int = 4 + 3 \* 3

2> my\_int = my\_int + 5

3> print(my\_int)

* 1. If you execute the three lines of code above, what will be printed?

Answer: 18

* 1. Explain your answer using the rules of assignment.

Answer: multiplication has precedence; hence the multiplication is executed first then adds the number 4. It follows the operator precedence rule: PEMDAS.

* 1. Rewrite line 2 using the += symbol Answer:

my\_int = 4 + 3 \* 3

my\_int += 5

print(my\_int)

1. Evaluate the following 3-line Python program:

1> my\_var1 = 13.0

2> my\_var2 = 4

3> print(my\_var1 % my\_var2)

If you execute this program, what will be printed and why?

Answer: 1.0, because the modulo operator returns the remainder of dividing the left-hand operand by the right-hand operand.