Fundamentals of Programming Python chapter 3, section 10 Exercises with Answers First Homework 2023/02/13

- 1. Is the literal 4 a valid python expression? Answer: Yes, literal 4 is a valid python expressoin.
- 2. Is the variable x a valid python expression? Answer: Yes, if the variable x exists, it is.
- 3. Is x + 4 a valid Python expression? Answer: Yes, it is; but x should be defined.
- 4. What affect does the unary + operator have when applied to a numeric expression? Answer: No effect all numeric values are positive by default.
- 5. Sort the following binary operators in order of high to low precedence: +, -, * , $^{\prime\prime}$, $^{\prime\prime}$, $^{\prime\prime}$, $^{\prime\prime}$, =.

Answer: The general order is PEMDAS.

parentheses, exponents, multiplication, division, addition and subtraction.

S0

6. Given the following assignment:

Indicate what each of the following Python statements would print.

Answers:

- (a) print("x") => x
- (b) print('x') => x
- (c) print(x) => 2
- (d) print("x + 1") => x + 1
- (e) print('x' + 1) => TypeError: can only concatenate str (not "int") to str
- (f) print(x + 1) => 3
- 7. Given the following assignments:
- i1 = 2
- i2 = 5
- i3 = -3
- d1 = 2.0
- d2 = 5.0
- d3 = -0.5

Evaluate each of the following Python expressions.

Answers:

- (a) i1 + i2 => 7
- (b) i1 / i2 => 0.4
- (c) i1 // i2 => 0
- (d) $i2 / i1 \Rightarrow 2.5$
- (e) i2 // i1 => 2
- (f) i1 * i3 => -6
- (g) d1 + d2 => 7.0
- (h) d1 / d2 => 0.4(i) d2 / d1 => 2.5
- (i) d2 + d1 = 2.3(j) d3 * d1 = -1.0
- (k) d1 + i2 => 4.0
- (l) i1 / d2 => 0.4
- (m) d2 / i1 => 2.5

```
(n) i2 / d1 => 2.5
(o) i1/i2*d1 => 0.8
(p) d1*i1/i2 => 0.8
(q) d1/d2*i1 => 0.8
(r) i1*d1/d2 => 0.8
(s) i2/i1*d1 => 5.0
(t) d1*i2/i1 => 5.0
(u) d2/d1*i1 => 5.0
(v) i1*d2/d1 => 5.0
8. What is printed by the following statement:
#print(5/3)
Answer: Nothing, because # converts the whole line to a comment.
9. Given the following assignments:
i1 = 2
i2 = 5
i3 = -3
d1 = 2.0
d2 = 5.0
d3 = -0.5
Evaluate each of the following Python expressions.
Answers:
(a) i1 + (i2 * i3)
                             => -13
(b) i1 * (i2 + i3)
                             => 4
(c) i1 / (i2 + i3)
                            => 1.0
(d) i1 // (i2 + i3)
                             => 1
(e) i1 / i2 + i3
                             => -2.6
(f) i1 // i2 + i3
                            => -3
                            => 8.6666666666666
(q) 3 + 4 + 5 / 3
(h) 3 + 4 + 5 // 3
                            => 8
                            => 4.0
(i) (3 + 4 + 5) / 3
(j) (3 + 4 + 5) // 3
                            => 4
(k) d1 + (d2 * d3)
                           => -0.5
(1) d1 + d2 * d3
                             => -0.5
                             => 0.9
(m) d1 / d2 - d3
(n) d1 / (d2 - d3)
                            => 0.36363636363636365
(o) d1 + d2 + d3 / 3
                            => 6.833333333333333
(p) (d1 + d2 + d3) / 3
                            => 2.166666666666665
(q) d1 + d2 + (d3 / 3)
                            => 6.833333333333333
(r) 3 * (d1 + d2) * (d1 - d3) => 52.5
10. What symbol signifies the beginning of a comment in Python?
Answer: it is Hashtag ('#').
11. How do Python comments end?
Answer: comments, are line elements, so if the line changes,
comment will end.
12. Which is better, too many comments or too few comments?
Answer: The happy medium is moderate amount of useful
comments. but, "when in doubt, add a remark" as the book explains itself...
13. What is the purpose of comments?
Answer: human readability:
in case a piece of code needs to be modified by another
programmer or even the same programmer, comments
```

aid them in reading and understanding the code faster.

14. Why is human readability such an important consideration? Answer: "Programmers are more important then programs" humans write code, so it is crucial that they understand the code easier and faster. 15. What circumstances can cause each of the following run-time errors to arise? Answers: => using undefined variable NameError => wrong value given to functions, like int('Omid') ValueError ZeroDivisionError => dividing by zero: 6/0 IndentationError => python declares blocks by indentation, so unnecessary indentation may cause this error OverflowError => math operations having very large results, example: 4.5**999999999999 SvntaxError => incomplete code, or problems regarding the way python should be written for example: print)(TypeError => trying to work with incompatible types for example print("your name is: " omid) Hint: Try some of following activities in the interpreter or within a Python program: print a variable that has not been assigned • convert the string 'two' to an integer • add an integer to a string • assign to a variable named end-point experiment adding spaces and tabs at various places in the code of an error-free Python program • compute raise a floating-point number to a large power, as in 1:510;000. 16. Consider the following program which contains some errors. You may assume that the comments within the program accurately describe the program's intended behavior. Answers: # Get two numbers from the user n1 = float(input()) # 1n2 = float(input()) # 2# Compute sum of the two numbers print(n1 + n2) # 3# Compute average of the two numbers print(n1+n2/2) # 4# Assign some variables d1 = d2 = 0 # 5=> both d1 and d2 are 0, d2 is useless. # Compute a quotient print(n1/d1) # 6=> zeroDivisionError since d1 is 0 # Compute a product n1*n2 = d1 # 7=> d1 = n1*n2# Print result print(d1) # 8 For each line listed in the comments, indicate whether or not an interpreter error, run-time exception,

or logic error is present. Not all lines contain an error.

```
17. Write the shortest way to express each of the following statements.
Answers:
(a) x = x + 1
                    => x += 1
(b) x = x / 2
                    => x /=2
(c) x = x - 1
                    => x -= 1
(d) x = x + y
                    => x += y
(e) x = x - (y + 7) \Rightarrow x - y + 7
                    => x *= 2
(f) x = 2*x
(g) number_of_closed_cases = number_of_closed_cases + 2*ncc
=> number_of_closed_cases += 2*ncc
18. What is printed by the following code fragment?
Answers:
x1 = 2
x2 = 2
x1 += 1
x2 -= 1
print(x1) => 3
print(x2) \Rightarrow 1
Why does the output appear as it does?
Aswer: Because x1 += 1 means x1 = x1 + 1 and x2 -= 1 means x2 = x2 - 1
19. Consider the following program that attempts to compute the circumference of a
circle given the
radius entered by the user. Given a circle's radius, r, the circle's circumference,
C is given by the
formula:
    C = 2pr
r = 0
PI = 3.14159
# Formula for the area of a circle given its radius
C = 2*PI*r
=> r is not defined yet.
# Get the radius from the user
r = float(input("Please enter the circle's radius: "))
 => should be above C = 2*PI*r
# Print the circumference
print("Circumference is", C)
(a) The program does not produce the intended result. Why?
 It's explained above.
(b) How can it be repaired so that it works correctly?
PI = 3.14159
r = float(input("Please enter the circle's radius: "))
C = 2*r*PI
print("Circumference is: ", C)
```