Advanced Programming: Homework-1

تمرین شماره یک درس برنامه سازی پیشرفته

3.10 Exercises

1. Is the literal 4 a valid Python expression?

-Yes

2. Is the variable x a valid Python expression?

-Yes

3. Is x + 4 a valid Python expression?

-Yes

4. What affect does the unary + operator have when applied to a numeric expression?

-The unary + operator is present only for completeness; when applied to a numeric value, variable, or expression, the resulting value is no different from the original value of its operand. For example : Omitting the unary + operator from the following statement

x = +y

does not change its behavior.

5. Sort the following binary operators in order of high to low precedence: +, -, *, //, /, %, =.

-In the table of priority and relationship of the following operators; Operators decrease in priority from top to bottom (that is, priority is higher to lower) and operators within a row have the same precedence.

Operators
* , / , // , %
+ , -
=

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)

(a) : x (b) : x (c) : 2 (d) : 3 (f) : 3

(e) : There is an error in this statement and the error is that 'x' is a string and the string cannot be added to a number.

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.

(a) i1 + i2 ➔ 2 + 5 = 7

(b) i1 / i2 ➔ 2 / 5 = 0.4

(c) i1 // i2 ➔ 2 // 5 = 0

(d) i2 / i1 ➔ 5 / 2 = 2.5

(e) i2 // i1 ➔ 5 // 2 = 2

(f) i1 * i3 ➔ 2 * (-3) = -6

(g) d1 + d2 ➔ 2.0 + 5.0 = 7.0

(h) d1 / d2 ➔ 2.0 / 5.0 = 0.4

(i) d2 / d1 ➔ 5.0 / 2.0 = 2.5

(j) d3 * d1 ➔ -0.5 * 2.0 = -0.1

(k) d1 + i2 ➔ 2.0 + 5 = 0.7

(l) i1 / d2 ➔ 2 / 5.0 = 0.4

(m) d2 / i1 ➔ 5.0 / 2 = 2.5

(n) i2 / d1 ➔ 5 / 2.0 = 2.5

(o) i1/i2*d1 ➔ 2 / 5 * 2.0 = 0.8

(p) d1*i1/i2 ➔ 2.0 * 2 / 5 = 0.8

(q) d1/d2*i1 ➔ 2.0 / 5.0 * 2 = 0.8

(r) i1*d1/d2 ➔ 2 * 2.0 / 5.0 = 0.8

(s) i2/i1*d1 ➔ 5 / 2 * 2.0 = 5.0

(t) d1*i2/i1 ➔ 2.0 * 5 / 2 = 5.0

(u) d2/d1*i1 ➔ 5.0 / 2.0 * 2 = 5.0

(v) i1*d2/d1 ➔ 2 * 5.0 / 2.0 = 5.0

8. What is printed by the following statement:

#print (5/3)

-This statement is a comment and nothing happens because there is a # sign before print and no command is executed and it is just an explanation, but if there was no # sign, the result would be 3.5, that is, 1.666 would be printed.

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.

(a) i1 + (i2 * i3) ➔ 2 + (5 *(-13)) = -13

(b) i1 * (i2 + i3) ➔ 2 * (5+ (-3)) = 4

(c) i1 / (i2 + i3) ➔ 2 / (5 + (-3)) = 1.0

(d) i1 // (i2 + i3) ➔ 2 // (5 + (-3)) = 1

(e) i1 / i2 + i3 ➔ 2 / 5 + (-3) = -2.6

(f) i1 // i2 + i3 ➔ 2 // 5 + (-3) = -3

(g) 3 + 4 + 5 / 3 ➔ 3 + 4 + 5 / 3 = 8.666666

(h) 3 + 4 + 5 // 3 ➔ 3 + 4 + 5 // 3 = 8

(i) (3 + 4 + 5) / 3 ➔ (3 + 4 + 5) / 3 = 4.0

(j) (3 + 4 + 5) // 3 ➔ (3 + 4 + 5) // 3 = 4

(k) d1 + (d2 * d3) ➔ 2.0 + (5.0 * (-0.5)) = -0.5

(l) d1 + d2 * d3 ➔ 2.0 + 5.0 * (-0.5) = -0.5

(m) d1 / d2 - d3 ➔ 2.0 / 5.0 - (-0.5) = 0.9

(n) d1 / (d2 - d3) ➔ 2.0 / (5.0 - (-0.5)) = 0.3636363636

(o) d1 + d2 + d3 / 3 ➔ 2.0 + 5.0 + (-0.5) / 3 = 6.833333333

(p) (d1 + d2 + d3) / 3 ➔ (2.0 + 5.0 + (-0.5)) / 3 = 2.16666666

(q) d1 + d2 + (d3 / 3) ➔ 2.0 + 5.0 + ((-0.5) / 3) = 6.833333333

(r) 3 * (d1 + d2) * (d1 - d3) ➔ 3 * (2.0 + 5.0) * (2.0 - (-0.5)) =52.2

10. What symbol signifies the beginning of a comment in Python?

- Comments in Python are beginig with a symbol, #,

11. How do Python comments end?

- It extends to the end of the line

12. Which is better, too many comments or too few comments?

- In my opinion, the more comments. The readability of the program increases, and even if a long time has passed since the program was

written or someone else wants to read the program, if there are many comments, you will quickly understand what program was written.

13. What is the purpose of comments?

- Increasing readability

- Explaining the code to others

- Understanding the code easily after a long-term

- Re-using the existing code (Helps remember why we used a specific command, method, or function in the code)

14. Why is human readability such an important consideration?

- Computer code is rarely written and forgotten. At a minimum you should review your own code before it is committed and ideally at least one other developer should look at it; Additionally while your development is fresh in your mind while you are writing the code it is likely you or some other programmer will have to read or modify the code at a later date. Keeping the code readable and adding an appropriate number of comments helps tremendously when the code has to be fixed or enhanced.

15. What circumstances can cause each of the following run-time errors to arise?

• NameError ➔ Raised when a variable is not found in the local or global scope.

• ValueError ➔ Raised when a function gets an argument of correct type but improper value.

• ZeroDivisionError ➔ Raised when the second operand of a division or module operation is zero.

• IndentationError ➔ Raised when there is an incorrect indentation.

• OverflowError ➔ Raised when the result of an arithmetic operation is too large to be represented.

• SyntaxError ➔ Raised by the parser when a syntax error is encountered.

• TypeError ➔ Raised when a function or operation is applied to an object of an incorrect type.

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.

Get two numbers from the user

n1 = float(input()) # 1 ➔ True

n2 = float(input()) # 2 ➔ True

Compute sum of the two numbers

print(n1 + n2) # 3 ➔ True

Compute average of the two numbers

print(n1+n2/2) # 4 ➔ False Because (n1+n2)must be added first and then divided by 2.

Assign some variables

d1 = d2 = 0 # 5 ➔ True

Compute a quotient

print(n1/d1) # 6 ➔ False Because d1 is zero, this expression (n1/d1) is wrong and gives the ZeroDivisionError.

Compute a product

n1*n2 = d1 # 7 ➔ False Because the variable is on the left :

d1= n1*n2

Print result

print(d1) # 8 ➔ True

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.

(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) ➔ x -= y + 7

(f) x = 2*x ➔ x *= 2

(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?

x1 = 2

x2 = 2

x1 += 1

```
x2 -= 1

print(x1)

print(x2)
```

Why does the output appear as it does?

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

# Get the radius from the user

r = float(input("Please enter the circle's radius: "))

# Print the circumference

print("Circumference is", C)
```

(a) The program does not produce the intended result. Why?

- Because our initial value r is zero and whatever we multiply, the formula becomes zero, so it does not give us any value or result.

(b) How can it be repaired so that it works correctly?

- Do not give an initial value or number

20. Write a Python program that ...

take 3 integer from the user shows the largest.

```
# to findmax

num1 = int(input("Please Enter Number 1: "))

num2 = int(input("Please Enter Number 2: "))

num3 = int(input("Please Enter Number 3: "))

result = max(num1,num2,num3)

print("Max number between %d, %d and %d is %d" %(num1, num2,num3, result))
```

```
Please Enter Number 1: 10

Please Enter Number 2: 50

Please Enter Number 3: 30

Max number between 10, 50 and 30 is 50
```

21. Write a Python program that ...

take 4 users and write the average numbers between the 4

```
# to find average

Number01 = int(input("Please Enter First Number: "))
```

```
Number02 = int(input("Please Enter Second Number: "))

Number03= int(input("Please Enter Third Number: "))

Number04= int(input("Please Enter Fourth Number: "))

Result = (Number01 + Number02 + Number03+ Number04) / 4

print ("Result :" (Result))
```

Please Enter First Number:10

Please Enter Second Number:5

Please Enter Third Number:11

Please Enter Fourth Number:2

Result:7