Chapter 3 Exercises

**1) Is the literal 4 a valid Python expression?**

Yes

**2) Is the variable x a valid Python expression?**

Yes, if the variable x exists, it is

**3) Is x + 4 a valid Python expression?**

If you have x initialised as a numeric datatype, yes otherwise no

44444What affect does the unary + operator have when applied to a numeric expression?444564165+asev4)**444444) What effect does the unary + operator have when applied to a numeric expression?**

Sum

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

= , \* , / , // , % , + , -

**6) Given the following assignment:**

**x = 2**

**Indicate what each of the following Python statements would print.**

**(a) print("x")** ---> x

**(b) print('x')** ---> x

**(c) print(x)** ---> 2

**(d) print("x + 1")** ---> x + 1

**(e) print('x' + 1)** ---> EROR : 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.**

**(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 = - 1.0

**(k) d1 + i2** ---> 2.0 + 5 = 7.0

**(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

**t) d1\*i2/i1** ---> 2.0 \* 5 / 2.0 = 5

**(u) d2/d1\*i1** ---> 5.0 / 2.0 \* 2 = 5

**(v) i1\*d2/d1** ---> 2 \* 5.0 / 2.0 = 5

**8) What is printed by the following statement:**

**#print(5/3)** ---> 1.6666666666666667

**9) Given the following assignment**

**1 = 2**

**2 = 5**

**3 = -3**

**1 = 2.0**

**2 = 5.0**

**3 = -0.5**

**valuate each of the following Python expressions.**

**(a) i1 + (i2 \* i3)** ---> 2 + (5 \* (-3)) = 2 + (-15) = -13

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

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

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

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

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

**(g) 3 + 4 + 5 / 3** ---> 7 + 5/3 = 1.6666666666666667 + 7 = 8.6666666666666667

**h) 3 + 4 + 5 // 3** ---> 7 + 1 = 8

**i) (3 + 4 + 5) / 3** ---> 12 / 3 = 4.0

**j) (3 + 4 + 5) // 3** ---> 12 // 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)) = 0.36363636363636365

**(o) d1 + d2 + d3 / 3** ---> 2.0 + 5.0 + (-0.5) / 3 = 6.833333333333333

**(p) (d1 + d2 + d3) / 3** ---> (2.0 + 5.0 + (-0.5)) / 3 = 2.1666666666666665

**(q) d1 + d2 + (d3 / 3)** ---> 2.0 + 5.0 + ((-0.5)/3) = 6.833333333333333

**(r) 3 \* (d1 + d2) \* (d1 - d3)** ---> 3 \* (2.0 + 5.0) \*(2.0 -(-0.5)) = 52.5

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

The symbol used to comment in Python is this symbol: ( # ) --> Sharp

**11) How do Python comments end?**

comments, are line elements, so if the line changes, comment will end.

**12) Which is better, too many comments or too few comments?**

Useful and brief

**13) What is the purpose of comments ?**

We use comments to explain python codes, to make code more readable and with comment we can describe our codes.

**14) Why is human readability such an important consideration?**

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?**

• **NameError**

Explain: The NameError occurs when you try to use a variable, function,

or module that doesn't exist or wasn't used in a valid way.

• **ValueError**

Explain: If Value Not Defined

• **ZeroDivisionError**

Explain: A ZeroDivisionError is raised when you try to divide by 0 .

This is part of the ArithmeticError Exception class.

• **IndentationError**

Explain :The indentation error can occur when the spaces or tabs are not placed properly.

• **ArithmeticError**

Explain :ArithmeticError is simply an error that occurs during numeric calculations.

ArithmeticError types in Python include: OverFlowError , ZeroDivisionError , FloatingPointError.

• **OverflowError**

Explain :An OverflowError exception is raised when an arithmetic operation exceeds the limits to be represented.

• **SyntaxError**

Explain : If the interpreter detects an invalid program statement during the translation phase,

it will terminate the program’s execution and report an error.

• **TypeError**

Explain :The Python TypeError is an exception that occurs when the data type of an object in an operation is inappropriate.

**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

n2 = 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

# Compute a quotient

print(n1/d1) # 6

# Compute a product

n1\*n2 = d1 # 7

# 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.**

**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)** ---> 3

**print(x2)** ---> 2

**Why does the output appear as it does?**

Because ( x1 += 1 == x1 = x1 + 1 ) AND ( x2 -= 1 == 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:**

**= 2pr**

**= 0**

**I = 3.14159**

**Formula for the area of a circle given its radius**

**= 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?**

=> 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)**

**20) Write a Python program that add two number with together.**

# This program adds two numbers

num1 = 1.5

num2 = 6.3

# Add two numbers

sum = num1 + num2

# Display the sum

print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))

output:

The sum of 1.5 and 6.3 is 7.8

**21) Write a Python program that calculate the area of a triangle.**

s = (a+b+c)/2

area = √(s(s-a)\*(s-b)\*(s-c))

# Python Program to find the area of triangle

a = 5

b = 6

c = 7

# Uncomment below to take inputs from the user

# a = float(input('Enter first side: '))

# b = float(input('Enter second side: '))

# c = float(input('Enter third side: '))

# calculate the semi-perimeter

s = (a + b + c) / 2

# calculate the area

area = (s\*(s-a)\*(s-b)\*(s-c)) \*\* 0.5

print('The area of the triangle is %0.2f' %area)

output:

The area of the triangle is 14.70