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

\* بله همه عبارات دارای ارزش هستند و کلمه 4 نیز جزو اینهاست ؛ مقدارهر متغیری ذخیره شده در محل حافظه محدود به همان متغیر است .

\* Yes, all phrases have value, and word 4 is one of them; The value of any variable stored in the memory location is limited to the same variable.

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

بله متغیر است یک عبارت پایتون معتبر برای نمونه هایی از پایتون معتبر که شناسایی می کند عبارتند از:

X.X2.total.port\_22

yes it is variable a valid python expression for examples of valid python identifies include: X.X2.total.port\_22

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

خیر، زیرا جمع فرعی یک عبارت معتبر در پایتون در نظر گرفته نمی شود.

No, because the subtotal is not considered a valid expression in Python.

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

وقتی این عملگر بین عبارات عددی قرار میگیرد آنها را با یکدیگر جمع میکند. \*

\* When this operator is placed between expressions, it adds them together.

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

1.\*,//,/,% 2.+,- 3.=

**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)** = error

\* 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** = 7

**(b) i1 / i2**  = 0.4

**(c) i1 // i2** = 0

**(d) i2 / i1** = 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

**(j) d3 \* d1** = - 1.0

**(k) d1 + i2** = 7.0

**(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)** = این دستور به علت کامنت شدن چیزی چاپ نمیکند \*

**\*** This command does not print anything because it is commented

**9. Given the following assignments:**

**i1 = 2**

**i2 = 5**

**i3 = -3**

**d1 = 2.0**

**d2 = 5.0**

**d3 = -0.5**

**d3 = -0.5**

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

**(a) i1 + (i2 \* i3)** = -13

**(b) i1 \* (i2 + i3)** = 4

**(c) i1 / (i2 + i3)** = 1

**(d) i1 // (i2 + i3)** = 1

**(e) i1 / i2 + i3** = - 2.6

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

**(g) 3 + 4 + 5 / 3** = 8.666666666666666

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

**(i) (3 + 4 + 5) / 3** = 0.4

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

**(k) d1 + (d2 \* d3)** = - 0.5

**(l) d1 + d2 \* d3** = - 0.5

**(m) d1 / d2 - d3** = 0.9

**(n) d1 / (d2 - d3)** = 0.363636363636365

**(o) d1 + d2 + d3 / 3** = 6.833333333333333

**(p) (d1 + d2 + d3) / 3** = 2.1666666666666665

**(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?** #

\* روش مرسوم هشتگ گذاری در ابتدای کامنت است که برای یک خط است ولی برای بیشتر از یک خط میتوانیم از رشته استفاده کنیم ولی آنرا نباید در قالب متغیر تعریف کنیم.

\* \* The conventional method is to use # at the beginning of the comment, which is for one line, but for more than one line, we can use the string, but it should not be defined in variable format.

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

. درواقع فقط تا پایان خط را ساپورت میکند و اگر میخواهیم کامنت ما بسته شود (تمام شود ) باید به خط بعد برویم # علامت \*

\* The # sign actually only supports till the end of the line ( finish ), and if we want to close the comment, we have to go to the next line.

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

\* هیچکدام ؛ کامنت نباید خیلی طولانی یا خیلی کوتاه باشد ، دراصل باید مختصرو مفید باشد و همچنین قابل فهم ... مطابق با منطق یک خطی تعریف شده در پایتون میتوان گفت استاندارد کامنت گذاشتن تاحد امکان همان یک خط است .

\* none; The comment should not be too long or too short, actually it should be brief and useful and also understandable... according to the logic of one line defined in Python, it can be said that the standard of commenting is one line.

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

\* در شرایطی که احساس میکنیم نیاز به توضیح راجبه بخشی از کد هست که اگر در ادامه نیاز به مراجعه به آن بخش بود متوجه نکات و موارد مهم ان بخش شویم ؛ توسط کامنت برای خود یادداشت میگذاریم و موارد مورد نیاز را توضیح میدهیم و نیاز به نگرانی راجبه این قسمت نیست چرا که این بخش در اجرای کد نقشی نخواهد داشت و شامل دستورهای کد نمیشود .

\* In a situation where we feel there is a need to explain a part of the code, because if you need to refer to it in the future, the important points and issues of that part are dealt with. I am reminding myself by note and i explains the required items and there is no need to worry about this part because this part does not play a role in the execution of the code and does not include the code commands.

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

\* چرا که سیستم ( کامپیوتر )چیزی نمیفهمد و درکنمیکند و این انسان است که باید دستورات را با منطق و زبان کامپیوتر به آن بفهماند تا کامپیوتر دستورات را اجرا کند .

\* Because the system (computer) does not understand anything and it is the human who must tell the commands with the logic and language of the computer to computer so that the computer executes the commands.

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

**1)nameError :**

1) استفاده از نام متغیر یا تابعی که هنوز تعریف نشده است .

2)غلط املایی نام متغیر یا نام تابع هنگام فراخوانی متغیر یا تابع .

3)استفاده از ماژول پایتون بدون وارد کردن ماژول و غیره .

1) Using the name of a change or a function that has not yet been defined.

2) Having a misspelling in the variable name or function name when calling the variable or function.

3) 3) Using the Python module without importing the module, etc.

**2)valueError :**

\* کاربر مقدار متغیری به یک تابع بدهد که اشتباه باشد و با تابع همخوانی نداشته باشد مثل دادن داده های str به float .

\* When the user gives a variable value to a function that is wrong and does not match the function, such as giving str data to float.

**3)zerodividionError :**

\* اگر عددی تقسیم بر صفر شود تعریف نشده است .

\* If a number is divided by zero, it is not defined.

**4)indentationError :**

\* هنگامی که در ساختار کد فرورفتگی های لازم انجام نشده باشد ؛ مثل فرورفتگی های ساختار if .

\* When the necessary indentations have not been made in the code structure; Like the indentations of the if structure.

**5)overflowError :**

\* مثل یک عملیات حسابی یا مقادیری که اندازه ی آنها و یا حاصل آنها بیشتر از مقدار تعریف شده ی هر متغیر باشد و سرریز مقادیری اتفاق بیوفتد .

\* Like an arithmetic operation or values, whose size or their result is greater than the defined value of each variable, and overflow of values ​​occurs.

**6)syntaxError :**

\* وقتی یک کد به مفسر میدهیم آنرا تجزیه میکند وآنرا اجرا میکند ؛اگر مفسر نتواند با موفقیت کد را ترجمه کند به این معناست که در جایی از کد خود از نحو نامعتبر استفاده شده است .

\* When we give a code to the interpreter, it parses it and executes it; if the interpreter cannot successfully translate the code, it means that invalid syntax has been used somewhere in the code.

**7)typeError :**

\* وقتی در پایتون یک برنامه مینویسیم و درآن رشته ((str )) را بایک عدد صحیح تقسیم کنیم ممکن است این خطا را بدهد.

\* When we write a program in Python and divide the string ((str)) by an integer, it may give this error.

**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 => print ((n1+n2)/2)**

**# Assign some variables**

**d1 = d2 = 0 # 5**

**# Compute a quotient**

**print(n1/d1) # 6=> d1=0 –then🡪 zerodivisionError**

**# Compute a product**

**n1\*n2 = d1 # 7=> we don’t know n1 and n2 so we are not sure result if the result is 0 or not … n1 or n2 must be 0 to get 0 in the result .**

**ما n1 و n2 را نمی دانیم بنابراین مطمئن نیستیم که نتیجه 0 است یا نه...n1 یا n2 باید 0 باشد تا 0 در نتیجه بدست بیاید.**

**# 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.**

\* خطا منطقی است نه مفسری => در هر سه مورد اینگونه است ... که دلیل آنها در قسمت بالا به طور کامل توضیح داده شده است.

\* The error is logical, not interpretive => it is like this in all three cases... the reason for which is fully explained in the above section.

**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 + z**

**(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)**

**result => for : print(x1) -> 3 & for : print(x2) -> 1**

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

\* چرا که قبل از اجرای دستور پرینت عملیات های محاسباتی روی آنها انجام میشود و جایگزین داده ها ی اولیه میشود و هنگام فراخوانی داده ها خروجی از داده های جدید استفاده میکند .

\* Because before executing the print command, calculation operations are performed on them and the new data replaces the initial data, and it uses the new data when it is called and in the output.

**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 = 2πr**

**r = 0**

**PI = 3.14159**

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

**r = float(input("Please enter the circle's 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?**

\* چون ابتدا دستور با شعاع 0 اجرا میشود و پس از انجام عملیات از ما شعاع جدید را میخواهد که دیگر روی محاسبات تاثیری ندارد و هرعددی ضربدر صفر برابر صفر است .

\* Because first the command is executed with a radius of 0, and after performing the operation, it asks us for the new radius, which no longer affects the calculations, and any number multiplied by zero is equal to zero.

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

\* خطا (ایراد کد) قرمز رنگ شده است و پاسخ درست با رنگ سبز در قسمت بالا نوشته شده است.

\* The error (code error) is colored red and the correct answer is written in green at the top.

When this operator is placed between expressions, it adds them together.

**20. Write a Python program that ...**

**ماکسیمم 3 عدد دلخواه کاربر را محسبه کند .**

**Calculate a maximum of 3 numbers as desired by the user.**

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))

**21. Write a Python program that ...**

**تعیین فرد یا زوج بودن عدد مورد نظر.**

**Determining whether the desired number is odd or even.**

**def odd\_even(num):**

**result = num % 2**

**if (result == 0):**

**print("%d is even!" %(num))**

**else:**

**print("%d is odd!" %(num))**

**num = int(input("Please Enter A Number: "))**

**odd\_even(num)**