

HW_week 1th_Bahman_#HOOMAN-ILEAYI#

1) Is the literal 4 a valid Python expression?

No, the literal 4 is not a valid Python expression. A valid Python expression would include an operator, such as `4 + 5` or `4 * 2`

2) Is the variable x a valid Python expression?

No, x is not a valid Python expression. It is a placeholder for a value that needs to be defined.

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

Yes, `x + 4` is a valid Python expression.

4) What effect does the unary `+` operator have when applied to a numeric expression?

The unary `+` operator has no effect when applied to a numeric expression. It is simply a placeholder that does nothing to the value of the expression.

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.

`print("x")`

(b) `print('x')`

(c) `print(x)`

(d) `print("x + 1")`

(e) `print('x' + 1)`

(f) `print(x + 1)`

"(a) `print("x")` - prints the string "x"

"(b) `print('x')` - prints the string "x"

(c) `print(x)` - prints the value 2

"(d) `print("x + 1")` - prints the string "x + 1"

(e) `print('x' + 1)` - produces an error as strings cannot be added to integers

(f) `print(x + 1)` - prints the value 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

(b) i1 / i2

(c) i1 // i2

(d) i2 / i1

(e) i2 // i1

(f) i1 * i3

(g) d1 + d2

(h) d1 / d2

(i) d2 / d1

(j) d3 * d1

(k) d1 + i2

(l) i1 / d2

(m) d2 / i1

(n) i2 / d1

(o) i1/i2*d1

(p) d1*i1/i2

(q) d1/d2*i1

(r) i1*d1/d2

(s) i2/i1*d1

(t) d1*i2/i1

(u) d2/d1*i1

(v) i1*d2/d1

(a) 7

(b) 0.4

(c) 0

(d) 2.5

(e) 5

(f) -6

(g) 7.0

(h) 0.4

(i) 2.5

(j) -1.0

(k) 7.0

(l) 0.4

(m) 2.5

(n) 5.0

(o) 1.0

(p) 1.0

(q) 0.4

(r) 0.4

(s) 2.5

(t) 10.0

(u) -3.0

(v) -2.5

8. What is printed by the following statement:

```
#print (5/3)
```

1.6666666666666667

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) $i1 + (i2 * i3)$

(b) $i1 * (i2 + i3)$

(c) $i1 / (i2 + i3)$

(d) $i1 // (i2 + i3)$

(e) $i1 / i2 + i3$

(f) $i1 // i2 + i3$

(g) $3 + 4 + 5 / 3$

(h) $3 + 4 + 5 // 3$

(i) $(3 + 4 + 5) / 3$

(j) $(3 + 4 + 5) // 3$

(k) $d1 + (d2 * d3)$

(l) $d1 + d2 * d3$

(m) $d1 / d2 - d3$

(n) $d1 / (d2 - d3)$

(o) $d1 + d2 + d3 / 3$

(p) $(d1 + d2 + d3) / 3$

(q) $d1 + d2 + (d3 / 3)$

(r) $3 * (d1 + d2) * (d1 - d3)$

(a) 7

(b) 17

(c) 0.2

(d) 0

- (e) 3.4
- (f) 4
- (g) 8.0
- (h) 8
- (i) 6.0
- (j) 6
- (k) 9.0
- (l) 11.5
- (m) 2.5
- (n)-2.5
- (o) 5.666666666666667
- (p) 5.666666666666667
- (q) 6.5
- (r) 18.0

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

#(HASHTAGS)

11. How do Python comments end?

Python comments end with a # symbol.

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

It depends on the context. Too many comments can make code difficult to read, while too few comments can make code difficult to understand. In general, it is best to strike a balance between the two and provide enough comments to explain the purpose of the code without cluttering it up.

13. What is the purpose of comments?

Comments are used to explain code, provide context, and make code easier to read and understand. They can also be used to document changes, alert other developers of potential issues, and provide helpful hints or tips.

14. Why is human readability such an important consideration?

Human readability is important because it allows people to easily understand the code. If the code is not readable, it can be difficult to debug and maintain. Additionally, if the code is not readable, it can be difficult for other developers to understand and use. Readable code also makes it easier for developers to collaborate on projects, as they can more easily understand each other's contributions.

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

- Name Error
- Value Error
- ZeroDivisionError
- Indentation Error
- Overflow Error
- Syntax Error
- Type Error

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 $1:510;000$.

.Name Error: Occurs when a variable is used that has not been assigned

Value Error: Occurs when a value is passed to a function that the function is not designed to handle

.ZeroDivisionError: Occurs when an attempt is made to divide by zero

Indentation Error: Occurs when incorrect indentation is used in the code of an error-free Python program

Overflow Error: Occurs when an operation produces a result that is too large to be stored in the available memory

.Syntax Error: Occurs when incorrect syntax is used in the code of an error-free Python program

Type Error: Occurs when an operation or function is applied to an object of an inappropriate type, such as trying to convert the string 'two' to an integer.

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.

- 1)no error
- 2)no error
- 3)no error
- 4)logic error
- 5)no error
- 6)runtime exception
- 7)syntax error
- 8)no error

17. Write the shortest way to express each of the following statements.

- (a) $x = x + 1$
 - (b) $x = x / 2$
 - (c) $x = x - 1$
 - (d) $x = x + y$
 - (e) $x = x - (y + 7)$
 - (f) $x = 2 * x$
 - (g) $\text{number_of_closed_cases} = \text{number_of_closed_cases} + 2 * \text{ncc}$
- (a) $x++$ (b) $x/=2$ (c) $x--$ (d) $x+=y$ (e) $x-=y+7$ (f) $x*=2$ (g) $\text{ncc}+=2*\text{ncc}$

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?

3 1

The output appears as it does because the += operator adds 1 to the value of x1, while the -= operator subtracts 1 from the value of x2

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:

$$C = 2\pi r$$

```
r = 0
PI = 3.14159
```

```
#
```

```
C = 2*PI*r
```

```
#
```

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

```
#
```

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

Formula for the area of a circle given its radius

Get the radius from the user

Print the circumference

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

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

(a) The program does not produce the intended result because it does not calculate the circumference of the circle using the formula provided

(b) The program can be repaired by adding a line of code to calculate the circumference using the formula provided: $C = 2 \cdot \pi \cdot r$. This line should be added before the print statement so that it is calculated before printing

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 =  $\sqrt{s(s-a)(s-b)(s-c)}$ 
```

```
# 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
```