Ai-Practical-Questions-Assignment

*Solved By : ARIB AHSAN*

April 15, 2025

1. **Write a Python program to display ‘Hello, AI World!’.**

Code:

*# Your answer here*

a

=

"

Hello,AI World!.

"

a

Output: 'Hello,AI World!.'

1. **Create a list of integers and print only the even numbers.**

Code:

*# Your answer here*

**for**

x

**in**

range

(

2

,

25

,

2

):

print

(

x

)

2

4

6

8

10

12

14

16

18

20

22

24

Code:

*# Your answer here*

ls

=

[

1

,

2

,

3

,

4

,

5

,

6

,

7

,

8

,

9

,

10

,

11

,

12

,

13

,

14

,

15

,

16

,

17

,

18

,

19

,

20

]

ls[

1

:

20

:

2

]

Output: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

**3. Write a function to calculate the factorial of a number.**

Code:

*# Your answer here*

num

=

int

(

input

(

"

Enter a number for find factorial:

"

))

factorial

=

1

*# check if the number is negative, positive or zero*

**if**

num

<

0

:

print

(

"

Sorry, factorial does not exist for negative numbers

"

)

**elif**

num

==

0

:

print

(

"

The factorial of 0 is 1

"

)

**else**

:

**for**

x

**in**

range

(

1

,num

+

1

):

factorial

=

factorial

\*

x

print

(

"

The factorial of

"

,num,

"

is

"

,factorial)

Enter a number for find factorial: 3

The factorial of 3 is 6

**4. Demonstrate the use of lambda to square a number.**

Code:

*# Your answer here*

x

=

int

(

input

(

"

Enter the number for squaring:

"

))

square

=

**lambda**

x: x

\*

\*

2

print

(

square(x

))

Enter the number for squaring: 5

25

Code:

*# Your answer here*

x

=

int

(

input

(

"

Enter the number for squaring:

"

))

square

=

x

\*

\*

2

print

(

f

"

Square of

**{**

x

**}**

is:

"

,square)

Enter the number for squaring: 2

Square of 2 is: 4

**5. Use map() to double the elements of a list.**

Code:

*# Your answer here*

numbers

=

[

1

,

2

,

3

,

4

,

5

]

*# Function to double a number*

**def**

double

(

x

):

**return**

x

\*

2

*# Using map() to double the elements of the list*

doubled\_numbers

=

list

(

map

(

double, numbers

))

print

(

"

Original numbers:

"

, numbers)

print

(

"

Doubled numbers:

"

, doubled\_numbers)

Original numbers: [1, 2, 3, 4, 5]

Doubled numbers: [2, 4, 6, 8, 10]

**6. Use filter() to get all the odd numbers from a list.**

Code:

*# Your answer here*

numbers

=

[

1

,

2

,

3

,

4

,

5

,

6

,

7

,

8

,

9

,

10

]

odd\_numbers

=

list

(

filter

(

**lambda**

x: x

%

2

!=

0

, numbers))

print

(

odd\_numbers

)

[1

, 3, 5, 7,

9]

**7. Write a program to read a string from the user and check if it is a palindrome.**

Code:

*# Your answer here*

x

=

str

(

input

(

"

Enter the sentence:

"

))

*# reverse the string*

rev\_str

=

reversed

(

x

)

*# check if the string is equal to its reverse*

**if**

list

(

x

)

==

list

(

rev\_str

):

print

(

"

The string is a palindrome.

"

)

**else**

:

print

(

"

The string is not a palindrome.

"

)

Enter the sentence: 121

The string is a palindrome.

**8. Create a dictionary with names as keys and ages as values. Print all keys having age > 20.**

Code:

*# Your answer here*

people

=

{

"

Alice

"

:

25

,

"

Bob

"

:

19

,

"

Charlie

"

:

22

,

"

David

"

:

18

,

"

Eve

"

:

30

}

*# Printing names of people with age greater than 20*

print

(

"

People with age greater than 20:

"

)

**for**

name, age

**in**

people

.

items():

**if**

age

>

20

:

print

(

name

)

People with age greater than 20:

Alice

Charlie

Eve

**9. Demonstrate file handling by writing and reading data from a text file.**

Code:

*# Your answer here*

z

=

open

(

"

arib ahsan.txt

"

,

"

a

"

)

z

.

write(

"

Now we can write anything in file

"

)

z

.

close()

Code:

z

=

open

(

"

arib ahsan.txt

"

,

"

r

"

)

print

(

z

.

read())

Now we can write anything in file

**10. Create a tuple and show how to access its elements.**

Code:

*# Your answer here*

tup

=

1

,

4

,

65

,

54

,

7

,

83

print

(

tup

)

tup[

2

:

4

]

(1

, 4, 65, 54,

7, 83)

[37]: (65, 54)

**11. Write a program using if-elif-else to check the grade of a student.**

Code:

*# Your answer here*

result

=

int

(

input

(

"

Enter the Percentage of your numbers:

"

))

**if**

result

>

=

90

:

grade

=

'

A+

'

**elif**

result

>

=

85

:

grade

=

'

B+

'

**elif**

result

>

=

75

:

grade

=

'

B

'

**elif**

result

>

=

65

:

grade

=

'

C+

'

**elif**

result

>

=

55

:

grade

=

'

C

'

**elif**

result

>

=

45

:

grade

=

'

D

'

**elif**

result

<

45

:

grade

=

'

F

'

print

(

"

Your Grade:

"

,grade)

Enter the Percentage of your numbers: 51

Your Grade: D

**12. Use a for loop to print numbers from 1 to 10 with their squares.**

Code:

*# Your answer here*

**for**

number

**in**

range

(

1

,

11

):

square

=

number

\*

\*

2

print

(

f

"

The square of

**{**

number

**}**

is

**{**

square

**}**

"

)

The square of 1 is 1

The square of 2 is 4

The square of 3 is 9

The square of 4 is 16

The square of 5 is 25

The square of 6 is 36

The square of 7 is 49

The square of 8 is 64

The square of 9 is 81

The square of 10 is 100

**13. Demonstrate list comprehension to create a list of cubes from 1 to 10.**

Code:

*# Your answer here*

**for**

number

**in**

range

(

1

,

11

):

cube

=

number

\*

\*

3

print

(

f

"

The cube of

**{**

number

**}**

is

**{**

cube

**}**

"

)

The cube of 1 is 1

The cube of 2 is 8

The cube of 3 is 27

The cube of 4 is 64

The cube of 5 is 125

The cube of 6 is 216

The cube of 7 is 343

The cube of 8 is 512

The cube of 9 is 729

The cube of 10 is 1000

**14. Write a nested function to calculate square of a number.**

Code:

*# Your answer here*

**def**

square\_calculator

():

**def**

square

(

n

):

**return**

n

\*

n

**return**

square

calculate\_square

=

square\_calculator()

number

=

5

result

=

calculate\_square(number)

print

(

f

"

The square of

**{**

number

**}**

is

**{**

result

**}**

.

"

)

The square of 5 is 25.

**15. Create a function with default parameters and call it with and without arguments.**

Code:

*# Your answer here*

**def**

greet

(

name

=

"

Guest

"

, greeting

=

"

Hello

"

):

**return**

f

"

**{**

greeting

**}**

,

**{**

name

**}**

!

"

*# Calling the function without arguments (uses default values)*

default\_greeting

=

greet()

print

(

default\_greeting

)

*# Output: Hello, Guest!*

*# Calling the function with one argument*

custom\_greeting

=

greet(

"

Alice

"

)

print

(

custom\_greeting

)

*# Output: Hello, Alice!*

*# Calling the function with both arguments*

personalized\_greeting

=

greet(

"

Bob

"

,

"

Welcome

"

)

print

(

personalized\_greeting

)

*# Output: Welcome, Bob!*

Hello, Guest!

Hello, Alice!

Welcome, Bob!

**16. Create a set and perform union and intersection with another set.**

Code:

set\_a

=

{

1

,

2

,

3

,

4

,

5

}

set\_b

=

{

4

,

5

,

6

,

7

,

8

}

*# Performing union*

union\_set = set\_a.union(set\_b) *# Using the union() method*

*# Alternatively, you can use the | operator*

*# union\_set = set\_a | set\_b*

*# Performing intersection*

intersection\_set = set\_a.intersection(set\_b) *# Using the intersection() method*

*# Alternatively, you can use the & operator*

*# intersection\_set = set\_a & set\_b*

*# Displaying the results* print("Set A:", set\_a) print("Set B:", set\_b)

print("Union of Set A and Set B:", union\_set) print("Intersection of Set A and Set B:", intersection\_set)

Set A: {1, 2, 3, 4, 5}

Set B: {4, 5, 6, 7, 8}

Union of Set A and Set B: {1, 2, 3, 4, 5, 6, 7, 8} Intersection of Set A and Set B: {4, 5}

**17. Demonstrate the use of try-except block to handle division by zero.**

Code:

*# Your answer here*

**def**

divide\_numbers

():

num1

=

int

(

input

(

"

Enter the num 1:

"

))

num2

=

int

(

input

(

"

Enter the num 2:

"

))

**try**

:

result

=

num1

/

num2

**except**

**ZeroDivisionError**

:

**return**

"

Error: Division by zero is not allowed

"

**else**

:

**return**

f

"

The result of

**{**

num1

**}**

divided by

**{**

num2

**}**

is

**{**

result

**}**

.

"

print

(

divide\_numbers

())

Enter the num 1: 22

Enter the num 2: 2

The result of 22 divided by 2 is 11.0.

**18. Create a class with a constructor and a method to display student details.**

Code:

*# Your answer here*

**class**

**Student**

:

**def**

\_\_init\_\_

(

self

, name, age, student\_id):

*"""Constructor to initialize student details."""*

self

.

name

=

name

self

.

age

=

age

self

.

student\_id

=

student\_id

**def**

display\_details

(

self

):

*"""Method to display student details."""*

print

(

f

"

Student Name:

**{**

self

.

name

**}**

"

)

print

(

f

"

Age:

**{**

self

.

age

**}**

"

)

print

(

f

"

Student ID:

**{**

self

.

student\_id

**}**

"

)

student1

=

Student(

"

Alice

"

,

20

,

"

S12345

"

)

student2

=

Student(

"

Bob

"

,

22

,

"

S67890

"

)

student1

.

display\_details()

print

()

*# Just for spacing*

student2

.

display\_details()

Student Name: Alice

Age: 20

Student ID: S12345

Student Name: Bob

Age: 22

Student ID: S67890

Age: 20

Student ID: S12345

Student Name: Bob

Age: 22

Student ID: S67890

**19. Demonstrate how to use enumerate in a loop.**

Code:

*# Your answer here*

*# Sample list of fruits*

fruits

=

[

"

apple

"

,

"

banana

"

,

"

cherry

"

,

"

date

"

]

*# Using enumerate in a for loop*

**for**

index, fruit

**in**

enumerate

(

fruits

):

print

(

f

"

Index:

**{**

index

**}**

, Fruit:

**{**

fruit

**}**

"

)

Index: 0, Fruit: apple

Index: 1, Fruit: banana

Index: 2, Fruit: cherry

Index: 3, Fruit: date

**20. Create a Python function that takes variable-length arguments and returns their sum.**

Code:

*# Your answer here*

**def**

sum\_of\_numbers

(

\*

args):

*"""Function to return the sum of variable-length arguments."""*

total

=

sum

(

args

)

*# Using the built-in sum function to calculate the total*

**return**

total

result1

=

sum\_of\_numbers(

1

,

2

,

3

,

4

,

5

)

print

(

f

"

The sum of 1, 2, 3, 4, 5 is:

**{**

result1

**}**

"

)

result2

=

sum\_of\_numbers(

10

,

20

,

30

)

print

(

f

"

The sum of 10, 20, 30 is:

**{**

result2

**}**

"

)

result3

=

sum\_of\_numbers()

*# No arguments*

print

(

f

"

The sum of no arguments is:

**{**

result3

**}**

"

)

The sum of 1, 2, 3, 4, 5 is: 15

The sum of 10, 20, 30 is: 60 The sum of no arguments is: 0