

HOMEWORK 02

Name it. Store it. Use it again.

A Python homework on variable assignment, valid names, multiple assignment, input(), and len().

```
name, age, group = "Aziza", 15, "Python-2"
```

STUDENT NAME

GROUP

DUE DATE

What you will practise

- ☐ Assign and update values
- ☐ Recognise valid variable names
- ☐ Use multiple assignment
- ☐ Read text with input()

How to complete it

- 1 Predict before running code.
- 2 Check your prediction in Python.
- 3 Correct mistakes clearly.
- 4 Finish the three required .py files.

Exact output matters.

Spaces, new lines, capital letters, and punctuation are part of the answer.

PART A

PREP - 12 POINTS

Variable assignment

Follow the statements in order. Write the exact output without running the code first.

1 One variable

```
x = 5
print(x)
```

Exact output:

2 Update a value

```
points = 10
points = points + 5
print(points)
```

Exact output:

3 A string value

```
city = "Tashkent"
print(city)
```

Exact output:

4 Copy, then change

```
x = 2
y = x
x = 8
print(y)
```

Exact output:

5 Different types

```
name = "Ali"
age = 15
print(name, age)
```

Exact output:

6 Store a length

```
word = "Python"
size = len(word)
print(size)
```

Exact output:

7. Write an assignment

Create a variable named score and assign the integer 27.

Python statement:

8. Think and explain

```
value = 10
value = 20
```

What value is stored now? What happened to the old value?

Your explanation:

PART B

PREP - 10 POINTS

The boring-but-important naming rules

For each name, circle VALID or INVALID. Then repair the invalid names.

1 student_name

VALID / INVALID

2 2students

VALID / INVALID

3 score2

VALID / INVALID

4 first-name

VALID / INVALID

5 _class

VALID / INVALID

6 class

VALID / INVALID

7 user name

VALID / INVALID

8 TOTAL

VALID / INVALID

9 myVariable

VALID / INVALID

10 len

VALID / INVALID

Repair these invalid names

2students _____

first-name _____

user name _____

class _____

Three syntax rules

1. One word only. 2. Letters, numbers, and underscore only. 3. Never begin with a number.

Note: len is valid syntax, but do not use it as a variable because it hides Python's len() function.

PART C

PREP - 12 POINTS

Multiple assignments

Match each name with one value. Predict the exact output.

1 Three values

```
x, y, z = 1, 2, 3
print(x, y, z)
```

Exact output:

2 Custom separator

```
x, y, z = 1, 2, 3
print(x, y, z, sep="-")
```

Exact output:

3 Mixed values

```
name, age, score = "Aziza", 15, 92.5
print(name, age, score)
```

Exact output:

4 Swap two values

```
a, b = 4, 9
a, b = b, a
print(a, b)
```

Exact output:

5 One value, three names

```
x = y = z = 7
print(x + y + z)
```

Exact output:

6 Count carefully

```
x, y = 1, 2, 3
```

RUNS / ERROR:

7. Complete the multiple assignment

Fill the blanks so each color is stored in the matching variable.

```
red, green, blue = _____, _____, _____
print(red, green, blue, sep=" | ")
```

Use these values: apple, leaf, sky

PART D

input() and len()

PREP - 12 POINTS

input() returns a string. len() counts every character, including spaces.

1. Type check

```
value = input() # user enters 42
```

Circle the type: string / integer / floating-point

2 Count a short name

```
name = input()
print(len(name))
```

Input: Ali - output:

3 Spaces count too

```
school = input()
print(len(school))
```

Input: Dombit School - output:

4 An empty input

```
text = input()
print(len(text))
```

Input: empty line - output:

5 Two inputs

```
first = input()
last = input()
print(first, last, sep="_")
```

Input: Ali / Vali - output:

6 Store the result

```
word = input()
length = len(word)
print("Length:", length)
```

Input: Python - output:

7. City and country

Read a city and country. Print both on one line.

Your code:

8. Nickname length

Read a nickname and print: Your name has N characters.

Your code:

PROGRAM 1

PROGRAM - 18 POINTS

Dombit Student Badge

Create a program named `student_badge.py`. Use variables so the output updates when input changes.

Starter code

```
name = input()
group = input()
school = "Dombit"
# Your print statements
```

Required output

```
Student: Aziza
Group: Python-2
School: Dombit
Name length: 5
```

Requirements

- ☐ Read name and group
- ☐ Assign school = "Dombit"
- ☐ Use `len(name)`
- ☐ Do not hard-code sample input

Write your code

Second test

```
Jasur
Python-3
```

Your result:

Hint: `input()` returns strings, and `len(name)` returns an integer.

PROGRAM 2

PROGRAM - 18 POINTS

Three-Word Unpacker

Create a program named `three_words.py`. Use variables so the output updates when input changes.

Starter code

```
first, second, third = input(), input(), input()
# Print the report below
```

Required output

```
Red | Green | Blue
```

Requirements

- ☐ Use one multiple assignment
- ☐ Call `input()` three times
- ☐ Use `sep=" | "`
- ☐ Print variables in order

Write your code

Second test

```
Python
Input
Length
```

Your result:

Hint: The number of names must match the number of values.

PROGRAM 3

PROGRAM - 18 POINTS

Secret Codename

Create secret_codename.py. Join three input values, store the result, and count its length.

Sample input

```
Ali  
Vali  
7
```

Required output

```
Codename: Ali_Vali_7  
Length: 10
```

Requirements

- ☐ Read first, last, and number
- ☐ Store the joined text in codename
- ☐ Use underscores between values
- ☐ Print len(codename)

Write your code

```
.....  
.....  
.....  
.....  
.....
```

Submission checklist

- ☐ student_badge.py runs
- ☐ three_words.py runs
- ☐ secret_codename.py runs
- ☐ Outputs match exactly

Points

Prep pages 2-5	46
Three programs	54
Total	100

Reflection

Which variable-name rule is easiest to forget? How do input() and len() work together?

Your answer:

```
.....
```