Exercises

1. Combining strings using concatenation

Create two string variables, 'greeting' and 'name', and concatenate them to print a personalized greeting.

▶ Code

Declare two more string variables, 'first_name' and 'last_name', and concatenate them to print the full name.

▶ Code

---

2. Arithmetic expressions in Python

Calculate and print the result of the expression: x = 10 + 5 * 2

▶ Code

Create a new variable 'y' and assign the result of the expression: y = 3 ** 2

▶ Code

---

3. Exponentiation in Python

Calculate and print the result of 2 raised to the power of 4 using the exponentiation operator.

▶ Code

---

4. Modulo operator in Python

Calculate and print the remainder when 15 is divided by 4 using the modulo operator.

▶ Code

---

5. Augmented assignment in Python

Initialize variables x and y with some values.

▶ Code

Use augmented assignment to add y to x and print the result.

▶ Code

---

6. Comments in Python

Add comments to the following code explaining the purpose of each line.

▶ Code

Print the result

▶ Code

---

7. Understanding data types - int vs float

Declare an integer variable 'integer_var' and a float variable 'float_var'.

▶ Code

Print the data type of each variable.

▶ Code

---

8. Multi-line string in Python

Create a multi-line string containing your address.

▶ Code

Print the multi-line string.

▶ Code

---

9. Booleans in Python

Declare boolean variables indicating weather conditions.

▶ Code

Print the values of the boolean variables.

▶ Code

---

10. Type error in Python

Fix the type error in the following statement and print the corrected string.

```
f1 = 0.25
f2 = 40.0
p = f1 * f2
bs = "The price is " + p
print(bs)
```

▶ Code

---

11. For loop with countries

Create a list of countries: Spain, France and Germany.

▶ Code

Use a for loop to print a message for each country.

▶ Code

---

12. For loop with numbers

Create a list of numbers: [1, 2, 3].

▶ Code

Use a for loop to calculate the product of all numbers and print the result.

▶ Code

print("Product of numbers:", total)

~~~
</details>
---

## 13. Return statements in Python

Define a function 'calculate_sum' that takes two parameters and returns
their sum.

<details><summary>Code</summary>
``` py
def calculate_sum(a, b):
    return a + b
~~~

Use the function to calculate and print the sum of 7 and 3.

▶ Code

Create a function 'is_positive' that takes a number as a parameter and returns True if it's positive, False otherwise.

▶ Code

Test the 'is_positive' function with both positive and negative numbers and print the results.

▶ Code
---

14. Defining a square function and calling it

Define a function 'square' that takes a number as a parameter and returns its square.

▶ Code

Use the 'square' function to calculate and print the square of 8.

▶ Code

Create a function 'calculate_area' that calculates the area of a square given its side length. Use the 'square' function to find the area of a square with side length 5.

▶ Code

area = calculate_area(5) print("The area of the square is:", area)

~~~
</details>
---

## 15. Using a for loop

Write a program that uses a for loop to print the numbers from 1 to 5.

<details><summary>Code</summary>
``` py
for i in range(1, 6):
    print(i)
~~~

16. Comparison operators in Python

Create a function 'compare_numbers' that takes two numbers as parameters and prints a message indicating which number is greater or if they are equal.

▶ Code

Test the 'compare_numbers' function with different pairs of numbers.

▶ Code
