

ASSIGNMENT-7.2

NAME: TEENESWARI

ROLLNO: 2303A51932

BATCH:30

TASK-1 PROMPT: num =

```
input("Enter a number: ") result
```

= num + 10 print(result) CODE:

```
File Edit Selection View Go ... 72.py x Training
EXPLORER TRAINING
72.py 75.py 72.py ...
1 num = int(input("Enter a number: "))
2 result = num + 10
3 print(result)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
29 PS C:\Users\neera\OneDrive\Desktop\Training> & C:/Users/neera/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/neera/OneDrive/Desktop/Training/72.py
Enter a number: 25
35
PS C:\Users\neera\OneDrive\Desktop\Training>

In 3, Col 14 Spaces: 4 UTF-8 CRLF {} Python 3.14.2 Go Live
30°C Sunny Search 14:28 10-02-2026
```

OBSERVATION:

The program was executed successfully. It accepts a number from the user using the input function. The entered value is added to 10 and the result is displayed. The output verifies the execution of input handling and addition operation in the program.

TASK-2

PROMPT:

```
def square(n):
```

```
    result = n * n
```

```
    return result
```

CODE:

The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree under the 'EXPLORER' tab, with a folder named 'TRAINING' containing numerous Python files like 72.py, 75.py, 123.py, etc. The main editor area shows the content of 72.py:

```
3
4 def square(n):
5     result = n * n
6     return result
```

The bottom status bar indicates the file is saved in 'master'. The terminal at the bottom shows the command PS C:\Users\neera\OneDrive\Desktop\Training> followed by two lines of output: 2 and PS C:\Users\neera\OneDrive\Desktop\Training> . The system tray shows the date and time as 10-02-2026.

OBSERVATION:

The function `square(n)` was executed successfully. It accepts a number as an argument, calculates its square by multiplying the number with itself, and returns the result. The function works correctly and produces the expected output.

TASK-3

PROMPT:

```
numbers = [10, 20, 30] for i in
```

```
range(0, len(numbers)):
```

```
    print(numbers[i])
```

CODE:

The screenshot shows a Windows desktop environment with the Visual Studio Code (VS Code) application open. The title bar reads "Training". The Explorer sidebar on the left lists several Python files under a "TRAINING" folder, including 72.py, 75.py, 123.py, decrement.py, factorial.py, ho.py, if else 1.py, if else basics.py, increment.py, list.py, palindrome.py, patterns.py, prime number.py, range.py, reverse of a number.py, reverse printing.py, strings.py, and sum of numbers.py. The 72.py file is selected. The main editor area contains the following code:

```
7
8     numbers = [10, 20, 30]
9     for i in range(0, len(numbers)+1):
10        print(numbers[i])
```

The terminal tab at the bottom shows the output of running the script:

```
PS C:\Users\neera\OneDrive\Desktop\Training> & C:/Users/neera/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/neera/OneDrive/Desktop/Training/72.py
20
20
20
20
```

The status bar at the bottom right indicates the date and time as "10-02-2026" and "14:31".

OBSERVATION:

The program was executed successfully. A list of numbers was created and a for loop was used to iterate through each element using index values. Each number in the list was printed as output. The results confirm that the loop and list operations are functioning correctly.

TASK-4

PROMPT:

total = 60 if

True:

```
print(total)
```

CODE:

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar has a tree view titled 'EXPLORER' under 'TRAINING' containing files like 72.py, 75.py, 123.py, decrement.py, factorial.py, ho.py, if else 1.py, if else basics.py, increment.py, list.py, palindrome.py, patternss.py, prime number.py, range.py, reverse of a number.py, reverse printing.py, strings.py, and sum of numbers.py. The main editor area shows the code for 72.py:

```
13 total += numbers[i]
14
15
16 print(total)
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\neera\OneDrive\Desktop\Training> & C:/Users/neera/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/neera/OneDrive/Desktop/Training/72.py
10
20
30
60
```

The status bar at the bottom right indicates the file is 3.14.2, the date is 10-02-2026, and the time is 14:32.

OBSERVATION:

The program was executed successfully. A variable was assigned a value and an if condition was used to check the statement. Since the condition is True, the value of the variable was printed. The output confirms the correct working of conditional statements in the program.

TASK-5

PROMPT: marks

= 85 if marks >=

90: grade = "A"

elif marks >= 80:

grade = "C" else:

grade = "B"

print(grade) CODE:

The screenshot shows a Visual Studio Code (VS Code) interface. The left sidebar has a 'TRAINING' folder containing several Python files: 72.py, 75.py, 123.py, decrement.py, factorial.py, ho.py, if else 1.py, if else basics.py, increment.py, list.py, palindrome.py, patternss.py, prime number.py, range.py, reverse of a number.py, reverse printing.py, strings.py, and sum of numbers.py. The main editor area displays the content of 72.py:

```
print(total)
marks = 85
if marks >= 90:
    grade = "A"
elif marks >= 80:
    grade = "B"
else:
    grade = "C"
print(grade)
```

The bottom terminal window shows the execution of the script and its output:

```
PS C:\Users\neera\OneDrive\Desktop\Training> & C:/Users/neera/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/neera/OneDrive/Desktop/Training/72.py
10
20
30
60
B
PS C:\Users\neera\OneDrive\Desktop\Training>
```

OBSERVATION:

The program was executed successfully. It assigns marks to a variable and uses conditional statements to determine the grade. Based on the given condition, the appropriate grade is selected and displayed. The output confirms the correct working of the if-elif-else control structure.