PYTHON LAB-3

Q: Program to print the give output:

```
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25
```

Program:

```
for i in range(1,6):
   for j in range(1,6):
      print(i*j,end=" ")
   print()
```

Objective

The goal of this program is to generate a multiplication pattern where each row contains the multiples of the row number.

Explanation of Code

1. **Outer Loop (for i in range(1, 6))**:
   - This loop runs from 1 to 5, representing the row numbers.
   - Each iteration represents a new row in the multiplication table.

2. **Inner Loop (for j in range(1, 6))**:

   This loop runs from 1 to 5, representing the column numbers.

   - It calculates the product of i * j, printing the result in the same row.
3. **print(i * j, end=" ")**:
   - The print() function prints the product of i and j on the same line.
   - The end=" " argument prevents the default newline after each print(), ensuring that numbers in a row are printed on the same line with spaces.
4. **print() (outside the inner loop)**:
   - This is executed after the inner loop completes.
   - It moves the cursor to the next line, starting a new row for the next iteration of the outer loop.

Concepts Used in the Program

Nested Loops

- A loop inside another loop is called a **nested loop**.
- The outer loop controls the number of rows, while the inner loop controls the number of columns.

Multiplication Table Logic

- Each number in the table is the product of its row index (i) and column index (j).

print() Function with end=" "

- The end=" " parameter ensures that the numbers in a row are printed on the same line with spaces instead of going to a new line.

Alternative Approach Using join()

Another way to achieve the same result without using end=" ":

```
for i in range(1, 6):
    print(" ".join(str(i * j) for j in range(1, 6)))
```

- Here, we use **list comprehension** to create a list of multiplication results.
- The " ".join() function joins the list elements into a single string, ensuring proper formatting.

Conclusion

This program effectively prints a multiplication pattern using nested loops. Understanding loops and print() behaviour is essential for controlling output formatting in Python.