24th September Notes

Functions in Python

- A **function** is a block of reusable code that performs a specific task.
- Functions make programs more **organized**, **readable**, and help **reduce repetition**.
- Functions are always represented with parentheses () when called.

◆ **Types of Functions**

1) Built-in Functions

Already available in Python.
Examples:

print()

len()

input()

- These functions are directly provided by Python and can be used without defining them.

2) User-Defined Functions

Functions created by the user using the def keyword.

General Syntax:

```
def fun_name(parameters):
   statements
   return value
```

- Here, fun_name is the function name, parameters are inputs, and the function can optionally return a value.

3) Lambda Function

```
lambda arguments: expression
```

- A **lambda function** is an anonymous (nameless) function that can be written in a single line using the lambda keyword.

💡 Example: Simple Interest

Formula:

SI=p×t×r100SI = \frac{p \times t \times r}{100}SI=100p×t×r

Where:

- **p** = Principal amount
- **t** = Time
- **r** = Rate of interest

📄 Function Examples

1) Function without input and without return

```
def si1():
  p = float(input("enter p value:"))
  t = float(input("enter t value:"))
  r = float(input("enter r value:"))
  si = (p * t * r) / 100
  print(f"the simple interest is {si} for pa={p}, time={t}, roi={r}")


si1()
```

Explanation:

- This function **does not take input arguments**.
- Instead, values are entered inside the function using input().
- The function **does not return** any value, it only prints the result.

2) Function with input and without return

```
def si2(p, t, r):
    si = (p * t * r) / 100
    print(f"the simple interest is {si} for pa={p}, time={t}, roi={r}")

# Direct call
si2(25000, 3, 2)

# With user input
x = float(input("enter p value:"))
y = float(input("enter t value:"))
z = float(input("enter r value:"))

si2(x, y, z)
```

Explanation:

- Here, the function **takes parameters** (p, t, r) as inputs.
- It **does not return** any value, only prints the result.
- The function can be called with direct values or user input.

3) Function without input and with return

```
def si3():
    p = float(input("enter p value:"))
    t = float(input("enter t value:"))
    r = float(input("enter r value:"))
    si = (p * t * r) / 100
    return si

var = si3()
print("the simple interest is:", var)
```

Explanation:

- The function **takes no input parameters**.
- It asks for values inside the function using input().
- Instead of printing, it **returns the calculated simple interest**.
- The result is stored in a variable (var) and printed outside.

✓ Returning multiple values (p, t, r, si):

```
def si3():
    p = float(input("enter p value:"))
    t = float(input("enter t value:"))
    r = float(input("enter r value:"))
    si = (p * t * r) / 100
    return (p, t, r, si)

var = si3()
print(f"the simple interest is {var[3]} for pa={var[0]}, time={var[1]}, roi={var[2]}")
```

Explanation:

- This function returns a **tuple** containing all values (p, t, r, si).
- Using var[index], we can print them individually.

4) Function with input and with return

```
def si4(p, t, r):
    si = (p * t * r) / 100
    return si

result = si4(40000, 2, 1)
print("the simple interest is:", result)
```

Explanation:

- This function **takes inputs** as parameters (p, t, r).
- It also **returns** the calculated simple interest.
- The result is stored in a variable and printed outside.