

Practical No 4

Aim: To implement user-defined functions with parameters, default arguments, return values, and demonstrate scope of variables.

Theory:

Functions help in **code reusability**, **modularity**, and **organization**. In Python, you can define functions using the def keyword.

User-Defined Functions

Syntax:

```
def function_name(parameters):  
    # function body  
    return value
```

Parameters and Arguments

- **Required parameters:** Must be passed.
- **Default arguments:** Have default values.

Example:

```
def greet(name="User"):  
    return f"Hello, {name}!"
```

Return Statement

Used to return a result from a function.

Example:

```
def add(a, b):  
    return a + b
```

Variable Scope

- **Local variables:** Declared inside a function, accessible only within that function.
- **Global variables:** Declared outside functions, accessible anywhere in the file.

Example:

```
x = 10 # Global  
def show():  
    x = 5 # Local  
    print("Inside:", x)  
  
show()  
print("Outside:", x)
```

```
def addition(a,b):
    sum=a+b
    return sum
print("Find the sum of two numbers")
x=int(input("Input first number:"))
y=int(input("Input second number:"))
sum=addition(x,y)
print("Sum=",sum)
```

Program

1. Write a Python function `calculate_area(length, width)` that takes two parameters and returns the area of a rectangle. Call the function with user input and display the result.

program:

```
def calculate_area(len,width):

    return len*width

print(f"area of rectangle is : {calculate_area(3,8)}")
```

output:

```
PS C:\Users\STUDENT\Desktop\py> python -u
"c:\Users\STUDENT\Desktop\py\area.py"
area of rectangle is : 24
```

2. Write a Python function `circle_area(radius)` that takes the radius of a circle as a parameter and returns its area. Use the formula:

program:

```
def calArea(radius):

    return 3.14*radius*radius

print(f"area of circle is: {calArea(4)}")
```

output:

```
PS C:\Users\STUDENT\Desktop\py> python -u  
"c:\Users\STUDENT\Desktop\py\areaofcircle.py"  
area of circle is: 50.24
```

3. Write a Python function `simple_interest(principal, rate, time)` that calculates and returns the simple interest using the formula:

program:

```
def cal_simple_interest(principle,rate,time):  
    return principle*rate*time/100  
  
print(f"simple inetrest is : {cal_simple_interest(3000,5,3)}")
```

output:

```
PS C:\Users\STUDENT\Desktop\py> python -u  
"c:\Users\STUDENT\Desktop\py\simple_interest.py"  
simple intrest is : 450.0
```

4. Write a Python function `celsius_to_fahrenheit(celsius)` that converts a given temperature from Celsius to Fahrenheit using the formula:

program:

```
def To_farr(cel):  
    farr=(cel*9/5)+32  
    return farr  
  
print(f"in farranide is: {To_farr(32)}")
```

output:

```
PS C:\Users\STUDENT\Desktop\py> python -u "c:\Users\STUDENT\Desktop\py\caltofah.py"  
in farranide is: 89.6
```

5. Write a Python function `is_even(number)` that takes a number as a parameter and returns `True` if the number is even, otherwise `False`. Accept a number from the user, call the function, and display whether the number is even or odd.

program:

```
def is_even(num):  
    if (num%2==0):  
        return True  
    else:  
        return False  
  
print(f"the number is even :{is_even(454)}")
```

```
PS C:\Users\STUDENT\Desktop\py> python -u  
"c:\Users\STUDENT\Desktop\py\evenOrOdd.py"  
the number is even :True
```