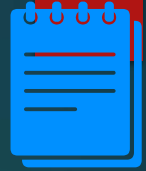




**Computer Science**

**Functions and procedures**



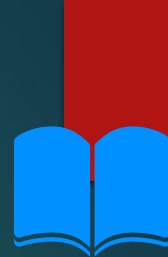
# Lesson Objectives

Students will learn:

- Application of functions (learnt in mathematics) in programming
- How are functions useful to programmers?
- What are procedures? How are they different from functions?

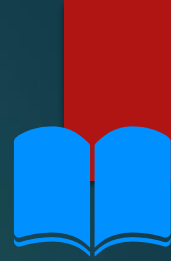


Content



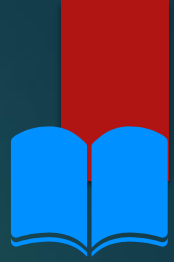
# Introduction

- Programs implement several tasks. Many times similar tasks are repeated.
- Typing the same code multiple times is time-consuming.
- Such tasks are written as modules such as functions, procedures or subprograms are written.
- These can be called multiple times in the main program reducing the time taken.



# Advantages of using functions and procedures

- The overall size of code is reduced as the codes are not repeated several times.
- Improves the readability of the program as only the name of these modules are used in the main program.
- Reduces the time taken to write and test the program. Repeated codes can be called from the main program and its enough if they are tested once.
- Reduces the time taken to debug the program. The code of functions and procedures are checked only once.



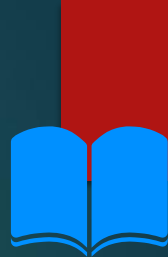
# Advantages of using functions and procedures

- In-built, pre-written and thoroughly checked functions:
  - ✓ These functions can be used by programmers.
  - ✓ This saves time and reduces the possibility of errors.
  - ✓ For example: In VB.net, function `InStr('Apple','ple')` returns the position of sub-string 'ple' in string 'Apple'. In this case, the `InStr` function returns the value 3.



# Function

- Functions are subprograms that can be called whenever required from the main program.



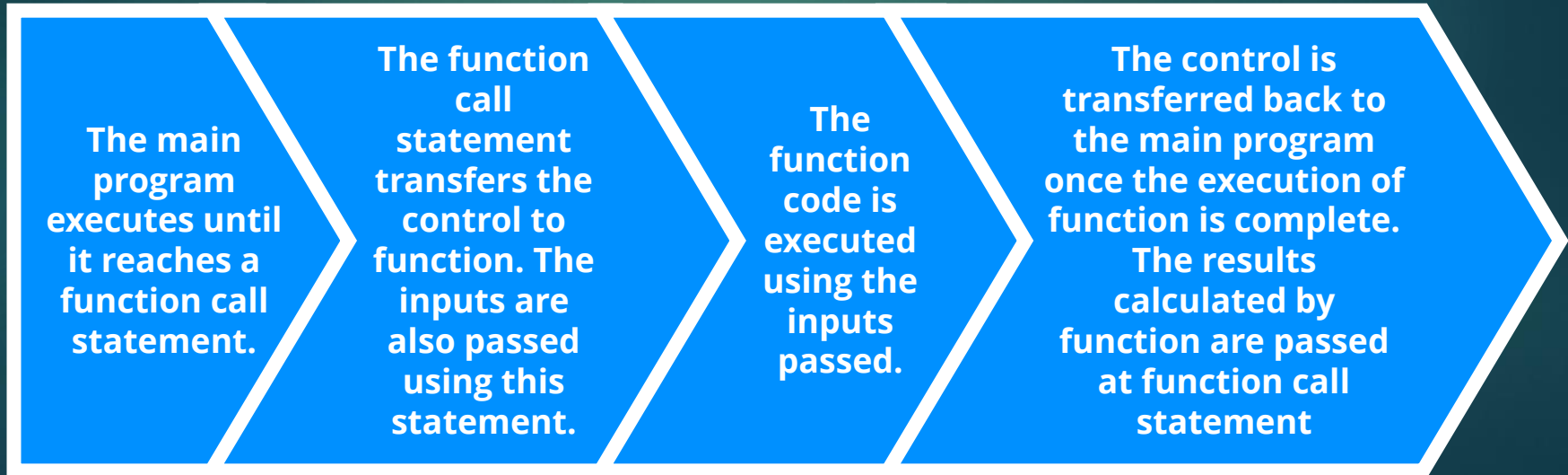
# Advantages of using functions

- The use of function reduces the time taken and improves the readability of the program.
- User need not know the function code. It's enough if the user knows how to use the function, that is, what inputs are given, what outputs are produced. This is an application of abstraction.





## Step-by-step process of a function call

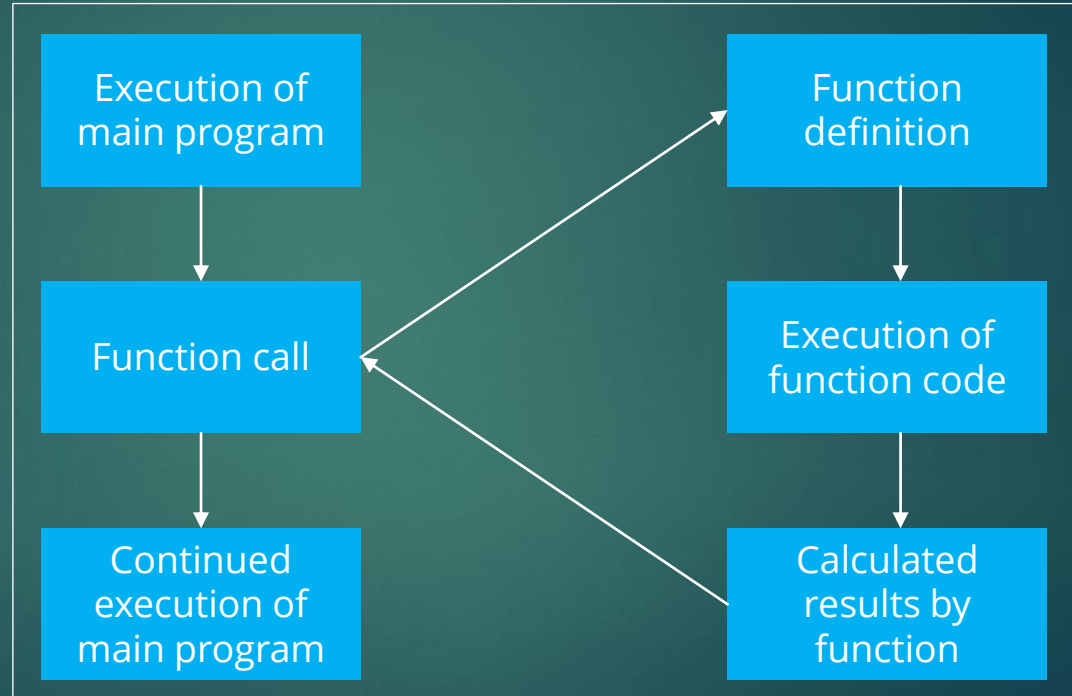




# Definition of a function

- The definition of a function requires:
  - ✓ Name of function
  - ✓ Inputs to be passed to the function
  - ✓ Function code
  - ✓ Outputs that are produced by the function

## Function call





## Example: Program to calculate volume of cylinder

- Mathematical formulae to calculate volume of cylinder is,  
Volume of cylinder=(area of base)× height
- The area of base is circle in cylinder.
- Let us use a function to calculate the area of circle in our program.



# Example: Program to calculate volume of cylinder

- Using a function to calculate the area of circle in our program.

Volume of cylinder=(area of circle)×  
height

```
#Function definition  
def area(radius):  
    area = 3.142*radius*radius  
    return area
```



# Example: Function call

```
#main program to find volume of cylinder  
radius_cyl=int(input('Enter the radius of cylinder: '))  
height_cyl=int(input('Enter the height of cylinder: '))  
volume_cylinder=area(radius_cyl)*height_cyl  
print("Volume of cylinder is: %.2f" %volume_cylinder)
```

```
#Function definition  
def area(radius):  
    area = 3.142*radius*radius  
    return area
```



# Example: Program to calculate volume of cylinder

- In this function:
  - ✓ Name of function: area
  - ✓ Input passed to function: radius
  - ✓ Function code:  $area = 3.142 \times radius \times radius$
  - ✓ Output from the function: area

```
#Function definition  
def area(radius):  
    area = 3.142*radius*radius  
    return area
```



# Example: Program to calculate volume of cone

- This function code can be reused to find the volume of cone too.

Volume of cone =  $\frac{1}{3} \times (\text{area of base}) \times \text{height}$

```
#Function definition
```

```
def area(radius):
```

```
    area = 3.142*radius*radius
```

```
    return area
```





## Example: Function call

```
#main program to find volume of cone  
radius_cone=int(input('Enter the radius of cone: '))  
height_cone=int(input('Enter the height of cone: '))  
volume_cone=1/3*area(radius_cone)*height_cone  
print("Volume of cone is: %.2f" %volume_cone)
```

```
#Function definition  
def area(radius):  
    area = 3.142*radius*radius  
    return area
```



## Example: Output

- Combining volume of cylinder and volume of cone in a same program and executing it results in this output.

```
Enter the radius of cylinder: 4
Enter the height of cylinder: 5
Volume of cylinder is: 251.36
Enter the radius of cone: 2
Enter the height of cone: 3
Volume of cone is: 12.57
```



# Procedures

- Similar to functions, procedures are sub-programs defined outside the main program.
- The difference between procedures and functions is procedures need not return a value but functions return a value.



## Example: Print text entered by user several times

```
#main program to print text  
string=str(input('Enter the text: '))  
times=int(input('Enter the number of times to  
print text: '))  
print_procedure(string,times)  
print('Execution of procedure is complete')
```

```
#Procedure definition  
def print_procedure(text, num):  
    for i in range(0,num):  
        print(text)
```



# Example: Print text entered by user several times

- The procedure 'print\_procedure' is defined and is called from the main program.
- It can be noted that the procedure does not return any value.
- Here it prints the text multiple times. Once, the execution of the procedure is complete, the control is passed to the main program.

```
#Procedure definition  
def print_procedure(text, num):  
    for i in range(0,num):  
        print(text)
```



## Example: Output

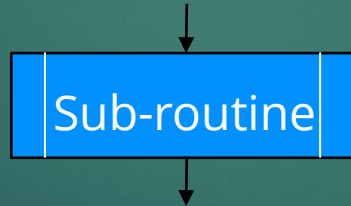
```
Enter the text: Hello World!!!  
Enter the number of times to print text: 6  
Hello World!!!  
Hello World!!!  
Hello World!!!  
Hello World!!!  
Hello World!!!  
Hello World!!!  
Execution of procedure is complete
```

```
#Procedure definition  
def print_procedure(text, num):  
    for i in range(0,num):  
        print(text)
```



# Sub-routines in flowchart

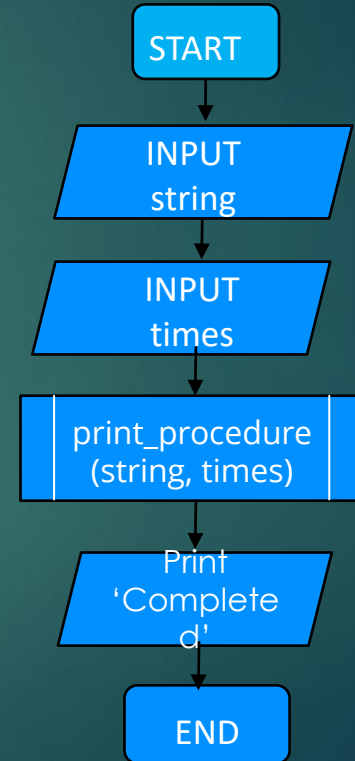
The flowchart symbol for representing a sub-routine is:





# Flowchart for sub-routines

```
#Procedure definition  
def print_procedure(text, num):  
    for i in range(0,num):  
        print(text)
```







# Global and local variables

- Variables and constants declared inside a function or procedure is local to that subroutine.
- To enable the usage of a variable throughout the program, it is made global using the keyword `global`.
- In this program, `x` is a global variable and `y` is a local variable to function `example`.
- The value of `x` can be accessed in the main program also but the value of `y` is undefined outside the function `example`

```
x = 9
def example():
    global x
    y = 5
    x = x * 2
    print(x)
    print(y)
example()
```



# Global and local variables

- Output:

```
18
5
```

```
x = 9
def example():
    global x
    y = 5
    x = x * 2
    print(x)
    print(y)
example()
```



# Global and local variables

- Executing the program to print the values of `x` and `y` in the main function,

```
18
5
18
Traceback (most recent call last):
  File "C:/Users/essuthan/Desktop/Python files/n1.py", line 12, in <module>
    print(y)
NameError: name 'y' is not defined
>>> |
```

```
x = 9
def example():
    global x
    y = 5
    x = x * 2
    print(x)
    print(y)
example()
print(x)
print(y)
```



# Let's review some concepts

## Functions

Functions are subprograms that can be called whenever required from the main program.

## Definition of a function:

Name of function

Inputs to be passed to the function

Function code

Outputs that are produced by the function

## Procedures

Similar to functions, procedures are subprograms defined outside the main program.

## Function call statement

Main program  $\xrightarrow{\text{inputs}}$  execution of function  $\xrightarrow{\text{output}}$  Main program

## Function vs. Procedures

Procedures need not return a value but functions return a value.



2.

Activity



# Activity-1

Duration: 10 minutes

1. You have created a Python program to find out whether a number is prime or not in topic “Basic programming constructs”. Now, implement the same program using function or procedure.
2. Which of the two: functions or procedures have you implemented? Why?



## Activity-2

Duration: 15 minutes

1. Create a python program to find the smallest of two numbers. Implement functions in your program.
2. Extend the program written above to find the smallest of three numbers.



3.

End of topic questions





## End of topic questions

1. What are the advantages of using functions and procedures in programs?
2. What are functions?
3. Explain the function-call process in detail.
4. What parameters are included in function definition?
5. How are procedures different from functions?
6. Using functions or procedures, write a Python program to check whether the given string is a palindrome or not.

# Credit

- TEACH COMPUTER SCIENCE