G.C.E. O/L Grade 11 ICT · Lesson 1

1.10 Sub-programs: Functions and Procedures

Syllabus 10.10. A **sub-program** is a small, named part of a program that does **one job**. We write it once and **call** (use) it whenever we need it – like a recipe for pol sambol that you can follow every day without rewriting it. Big programs are built from many small sub-programs.

1. Why use sub-programs?

Reusability
Write once, call many times. No need to copy the same code again.

Readability
Names like PrintLine and GetGrade make the main program easy to understand.

Maintainability
To fix or change a job, change it in one place only.

Teamwork and testing
Different people can write and test different sub-programs.

2. Two kinds of sub-programs

	Value-returning	Non-value-returning
Pascal name	function	procedure
Python	def ... with return value	def ... without return value
Gives back	one value (e.g. a grade)	nothing – it just does a job
Example	GetGrade(82) → 'A'	PrintLine

3. Parameters and arguments

Values we send into a sub-program are **parameters**. In area(length, width), length and width are parameters. When we call area(5, 4), the values 5 and 4 are the **arguments**.

4. A value-returning function

GetGrade receives a mark and **returns** a grade letter, using the O/L grade limits (A 75, B 65, C 50, S 35).

Pseudocode

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

```
FUNCTION GetGrade(Marks)
  IF Marks >= 75 THEN RETURN "A"
  ELSE IF Marks >= 65 THEN RETURN "B"
  ELSE IF Marks >= 50 THEN RETURN "C"
  ELSE IF Marks >= 35 THEN RETURN "S"
  ELSE RETURN "W"
  ENDIF
ENDFUNCTION

BEGIN
  FOR k = 1 TO 3
    INPUT M
    DISPLAY GetGrade(M)
  ENDFOR
END
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Value-returning sub-program (function): GetGrade

Pascal

```
program GradeFunction;
var
  k, m: integer;

function GetGrade(marks: integer): char;
begin
  if marks >= 75 then
    GetGrade := 'A'
  else if marks >= 65 then
    GetGrade := 'B'
  else if marks >= 50 then
    GetGrade := 'C'
  else if marks >= 35 then
    GetGrade := 'S'
  else
    GetGrade := 'W';
end;

begin
  for k := 1 to 3 do
  begin
    write('Enter mark: ');
    readln(m);
    writeln('Grade: ', GetGrade(m));
  end;
end.
```

Python

```
def get_grade(marks):
    if marks >= 75:
        return "A"
    elif marks >= 65:
        return "B"
    elif marks >= 50:
        return "C"
    elif marks >= 35:
        return "S"
    else:
        return "W"

for k in range(3):
    m = int(input("Enter mark: "))
    print("Grade:", get_grade(m))
```

Sample run

```
Enter mark: 82
Grade: A
Enter mark: 47
Grade: S
Enter mark: 66
Grade: B
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

> 💡 In Pascal, the result is given by assigning to the function name: GetGrade := 'A'. In Python we write return "A".

5. Procedures (non-value-returning)

A shop bill uses PrintLine three times and ShowItem for each item. PrintLine has **no parameters**; ShowItem has three.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Non-value-returning sub-programs (procedures): a shop bill

Pascal

```
program BillProcedures;

procedure PrintLine;
begin
  writeln('------------------------');
end;

procedure ShowItem(item: string; price, qty: integer);
begin
  writeln(item, ': Rs. ', price * qty);
end;

begin
  PrintLine;
  writeln('SAMPLE SHOP BILL');
  PrintLine;
  ShowItem('Bread', 120, 2);
  ShowItem('Milk packet', 280, 1);
  ShowItem('Eggs', 55, 6);
  PrintLine;
end.
```

Python

```
def print_line():
    print("-" * 24)

def show_item(item, price, qty):
    print(item + ": Rs.", price * qty)

print_line()
print("SAMPLE SHOP BILL")
print_line()
show_item("Bread", 120, 2)
show_item("Milk packet", 280, 1)
show_item("Eggs", 55, 6)
print_line()
```

Sample run

```
------------------------
SAMPLE SHOP BILL
------------------------
Bread: Rs. 240
Milk packet: Rs. 280
Eggs: Rs. 330
------------------------
```

6. Calling a function many times

A function can be used inside an expression, just like a number.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

A function with two parameters: area of a rectangle

Pascal

```
program RectArea;

function Area(length, width: integer): integer;
begin
  Area := length * width;
end;

begin
  writeln('Room 1 = ', Area(5, 4));
  writeln('Room 2 = ', Area(6, 3));
  writeln('Both = ', Area(5, 4) + Area(6, 3));
end.
```

Python

```
def area(length, width):
    return length * width

print("Room 1 =", area(5, 4))
print("Room 2 =", area(6, 3))
print("Both =", area(5, 4) + area(6, 3))
```

Sample run

```
Room 1 = 20
Room 2 = 18
Both = 38
```

💡 Built-in sub-programs you already use: writeln, readln, sqrt (Pascal); print, input, len, round (Python).

Key Words

Sub-program – A named part of a program that does one job.

Function – A sub-program that returns a value.

Procedure – A sub-program that does a job but returns no value.

Call – Using a sub-program by writing its name.

Parameter – A variable that receives a value sent into a sub-program.

Argument – The actual value sent when calling, e.g. 5 in area(5, 4).

Return value – The result a function gives back.

Reusability – Writing code once and using it many times.

Readability – How easy code is to read and understand.

Maintainability – How easy code is to fix and change.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

def – Python keyword to define a sub-program.

Built-in function – A function already provided by the language, e.g. len.

Sample Q&A

1. What is a sub-program?

> **Answer:** A named part of a program that does one job and can be called when needed.

2. Give three advantages of sub-programs.

> **Answer:** Reusability, readability and maintainability.

3. Difference between a function and a procedure?

> **Answer:** A function returns a value; a procedure does not.

4. What is GetGrade(66)?

> **Answer:** B.

5. What does area(6, 3) return?

> **Answer:** 18.

6. In area(5, 4), what are 5 and 4?

> **Answer:** Arguments (values for the parameters length and width).

MCQ

1. A named part of a program that does one job is a…

A. data type
B. comment
C. constant
D. sub-program

2. A sub-program that returns a value is a…

A. function
B. array
C. procedure
D. loop

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. In Pascal, a sub-program that returns no value is a…

A. record
B. const
C. procedure
D. function

4. “Write once, use many times” describes…

A. nesting
B. compiling
C. reusability
D. debugging

5. Fixing a job in one place only shows…

A. maintainability
B. iteration
C. random access
D. precedence

6. Clear names such as PrintLine improve…

A. readability
B. speed of the printer
C. the data type
D. memory size

7. GetGrade(82) returns…

A. S
B. B
C. C
D. A

8. GetGrade(47) returns…

A. C
B. S
C. W
D. B

9. area(5, 4) + area(6, 3) equals…

A. 18
B. 20
C. 9
D. 38

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

10. In def area(length, width):, length and width are...

A. reserved words
B. parameters
C. arrays
D. constants

11. The Python keyword that defines a sub-program is...

A. func
B. proc
C. def
D. sub

12. In Python, a function gives back its result with...

A. break
B. return
C. input
D. print

13. In Pascal, a function returns a value by...

A. assigning to the function name
B. using print
C. writing end.
D. using readln

14. ShowItem('Eggs', 55, 6) displays...

A. Eggs: Rs. 55
B. Eggs: Rs. 330
C. Eggs: Rs. 6
D. Eggs: Rs. 61

15. Which is a built-in Python function?

A. PrintLine
B. ShowItem
C. len
D. GetGrade

Structured Questions

1. State three advantages of using sub-programs. (3 marks) *(3 marks)*

> Reusability – write once, call many times; readability – easy to understand; maintainability – change in one place (1 each).

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. Distinguish between a value-returning and a non-value-returning sub-program with an example of each. (4 marks) *(4 marks)*

> Value-returning (function) gives back a value, e.g. GetGrade (2); non-value-returning (procedure) only does a task, e.g. PrintLine (2).

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

3. Write a function (Pascal or Python) that receives a mark and returns the O/L grade A, B, C, S or W. (5 marks) *(5 marks)*

Function heading with parameter (1); IF ≥ 75 A (1); ELSE IF ≥ 65 B, ≥ 50 C (1); ≥ 35 S else W (1); return / assign result (1).

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Value-returning sub-program (function): GetGrade

Pascal

```
program GradeFunction;
var
  k, m: integer;

function GetGrade(marks: integer): char;
begin
  if marks >= 75 then
    GetGrade := 'A'
  else if marks >= 65 then
    GetGrade := 'B'
  else if marks >= 50 then
    GetGrade := 'C'
  else if marks >= 35 then
    GetGrade := 'S'
  else
    GetGrade := 'W';
end;

begin
  for k := 1 to 3 do
  begin
    write('Enter mark: ');
    readln(m);
    writeln('Grade: ', GetGrade(m));
  end;
end.
```

Python

```
def get_grade(marks):
    if marks >= 75:
        return "A"
    elif marks >= 65:
        return "B"
    elif marks >= 50:
        return "C"
    elif marks >= 35:
        return "S"
    else:
        return "W"

for k in range(3):
    m = int(input("Enter mark: "))
    print("Grade:", get_grade(m))
```

Sample run

```
Enter mark: 82
Grade: A
Enter mark: 47
Grade: S
Enter mark: 66
Grade: B
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

4. def area(length, width): return length * width
What is displayed by print(area(5, 4) + area(6, 3))? Name the parameters and arguments. (3 marks) *(3 marks)*

> 38 (1); parameters: length, width (1); arguments: 5, 4, 6, 3 (1).

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

5. Write a procedure ShowItem that receives an item name, price and quantity and displays the item total. Show one call. (4 marks) *(4 marks)*

Procedure heading with three parameters (1); calculate price × qty (1); display (1); call e.g. ShowItem('Bread', 120, 2) (1).

Non-value-returning sub-programs (procedures): a shop bill

Pascal

```
program BillProcedures;

procedure PrintLine;
begin
  writeln('------------------------');
end;

procedure ShowItem(item: string; price, qty: integer);
begin
  writeln(item, ': Rs. ', price * qty);
end;

begin
  PrintLine;
  writeln('SAMPLE SHOP BILL');
  PrintLine;
  ShowItem('Bread', 120, 2);
  ShowItem('Milk packet', 280, 1);
  ShowItem('Eggs', 55, 6);
  PrintLine;
end.
```

Python

```
def print_line():
    print("-" * 24)

def show_item(item, price, qty):
    print(item + ": Rs.", price * qty)

print_line()
print("SAMPLE SHOP BILL")
print_line()
show_item("Bread", 120, 2)
show_item("Milk packet", 280, 1)
show_item("Eggs", 55, 6)
print_line()
```

Sample run

```
------------------------
SAMPLE SHOP BILL
------------------------
Bread: Rs. 240
Milk packet: Rs. 280
Eggs: Rs. 330
------------------------
```

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Summary

- Sub-program = named part doing one job.
- Benefits: reusability, readability, maintainability.
- Function returns a value.
- Procedure returns no value.
- Parameters receive; arguments are sent.
- Python: def … return.
- Pascal: FunctionName := value.
- Built-ins: print, len, writeln, sqrt.

MCQ answer key – 1.10

1. **D** – Sub-program.
2. **A** – Function.
3. **C** – Procedure.
4. **C** – Reusability.
5. **A** – Maintainability.
6. **A** – Readability.
7. **D** – 82 ≥ 75.
8. **B** – 35 ≤ 47 < 50.
9. **D** – 20 + 18.
10. **B** – Parameters.
11. **C** – def.
12. **B** – return.
13. **A** – GetGrade := 'A'.
14. **B** – 55 × 6.
15. **C** – Built-in.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.