Functions and procedures

Algorithms and programming (Paper 2) · revision notes

Sub programs (procedures and functions) are named blocks of code that do one task and can be called from anywhere in a program. You need the difference between a procedure and a function, how parameters and return values work, local and global variables, and why sub programs make code better. Expect definitions and short functions to write.

Key facts

- Sub program: named block of code for one task, run by calling its name
- Function: returns a value (function … return … endfunction)
- Local variable: exists only inside its sub program
- Benefits: reuse, shorter code, easier to test and maintain, shared between a team
- Procedure: does a task, returns no value (procedure … endprocedure)
- Parameter: variable in the sub program's first line that receives a value passed in
- Global variable: can be used anywhere (global in ERL)

Notes

Procedures and functions

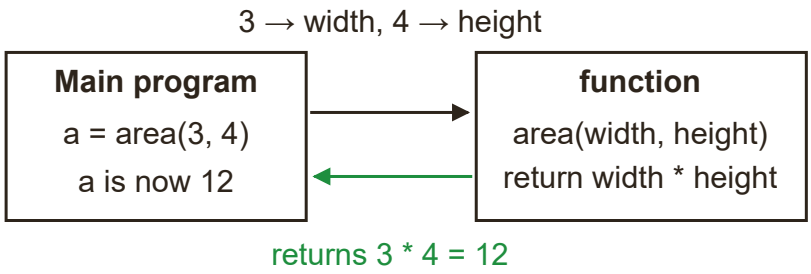
- A **sub program** is a named block of code that carries out a specific task. It runs when it is **called** by its name.
- A **procedure** carries out a task but does not return a value.

```
procedure greet(name)
   print("Hello " + name)
endprocedure

greet("Sam")
```

- A **function** carries out a task and **returns** a value to the code that called it.

```
function area(width, height)
   return width * height
endfunction

a = area(3, 4)
```

- Here a becomes 12. The returned value is usually stored, output or used in a calculation.



A function call: values go in through the parameters, and one value comes back

Parameters and variables

- **Parameters** are the variables in brackets in the sub program's first line (width and height). They receive the values passed in when it is called (3 and 4). An array can be passed in as a parameter, and a function can return an array.
- A **local variable** is created inside a sub program. It only exists while the sub program runs and cannot be used outside it.
- A **global variable** is declared in the main program and can be used anywhere in the program. In ERL: `global total = 0`. Constants can also be local or global in the same way.
- Local variables are preferred: the same name can be used in different sub programs without clashing, and a sub program cannot change data elsewhere by mistake.

Why use sub programs

- The code is written once and can be called many times, so the program is shorter.
- Each sub program can be tested on its own, so errors are easier to find.
- A change only needs making in one place, so the program is easier to maintain.
- Different programmers can write different sub programs, and sub programs can be reused in other programs.

How to answer each type of question

State the difference between a function and a procedure 2 marks **Grade 5**

1. Function: returns a value.
2. Procedure: does not return a value.
3. Two sentences, one about each.

Example. State the difference between a function and a procedure. [2]

Model answer. A function returns a value to the part of the program that called it (1). A procedure does not return a value (1).

Identify parts of a sub program

3 to 4 marks **Grade 6**

1. Parameters are in brackets in the first line.
2. Local variables are assigned inside the sub program.
3. Trace the call to find the value returned or output.

Example. Look at the code.

```
01 function largest(a, b)
02    if a > b then
03       big = a
04    else
05       big = b
06    endif
07    return big
08 endfunction
09 result = largest(7, 12)
10 print(result)
```

(a) Identify the parameters of the function. [1]
(b) Identify a local variable. [1]
(c) State the output of the program. [1]
(d) Give **one** benefit of writing this code as a function. [1]

Model answer.

(a) a and b (1)
(b) big (1)
(c) 12 (1)
(d) It can be called many times without writing the code again / it can be tested on its own (1)

Write a function

3 to 4 marks | **Grade 7**

1. Write the first line with the function name and its parameters, exactly as the question names them.
2. Do the processing using the parameters (not new input).
3. Return the result, and end with endfunction.
4. If asked, show how to call it and use the value returned.

Example. (a) Write a function called `addVAT` that takes a price as a parameter and returns the price with 20% VAT added. [3]
(b) Write a line of code that uses the function to output the price of an item costing £15. [1]

Model answer.

(a)

```
function addVAT(price)
   total = price * 1.2
   return total
endfunction
```

First line with the name addVAT and the parameter price (1); calculates price × 1.2 (1); returns the result (1).

(b) `print(addVAT(15))` (1)

Don’t lose marks

- Saying a procedure 'returns nothing useful' without saying a function returns a value.
- Using input() inside a function instead of using the parameters passed in.
- Printing the answer inside a function when the question asks for it to be returned.
- Calling a function but not storing or using the value it returns.
- Using a local variable outside the sub program it belongs to.

More tips

Memory tricks

- Function = gives you something back (like a vending machine). Procedure = just does a job (like a light switch).
- If you see return, it is a function.
- Local = lives inside the sub program; global = visible everywhere.

Exam technique

- Use exactly the sub program name and parameter names the question gives.
- When asked for benefits, give different ones (reuse, testing, maintenance, teamwork), each with a reason.
- 'Return' and 'output' are different: return sends a value back to the code; output shows it on screen.

What each grade needs

What you need to be able to do, from the first marks up to the top grade.

Grade 4

Describe what a sub program is

A named block of code that does a specific task and can be called when needed.

Grade 5

Tell functions and procedures apart

A function returns a value; a procedure does not.

Grade 5

Give benefits of using sub programs

Code is reused, easier to test and maintain, and can be shared between programmers.

Grade 6

Identify parameters and return values

Parameters receive values passed in; return sends a value back to where it was called.

Grade 7

Write a function with parameters

Use function … endfunction, with parameters and a return statement.

Grade 7

Explain local and global variables

A local variable exists only in its sub program; a global variable can be used anywhere.

Free notes, Full packs, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers. Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.