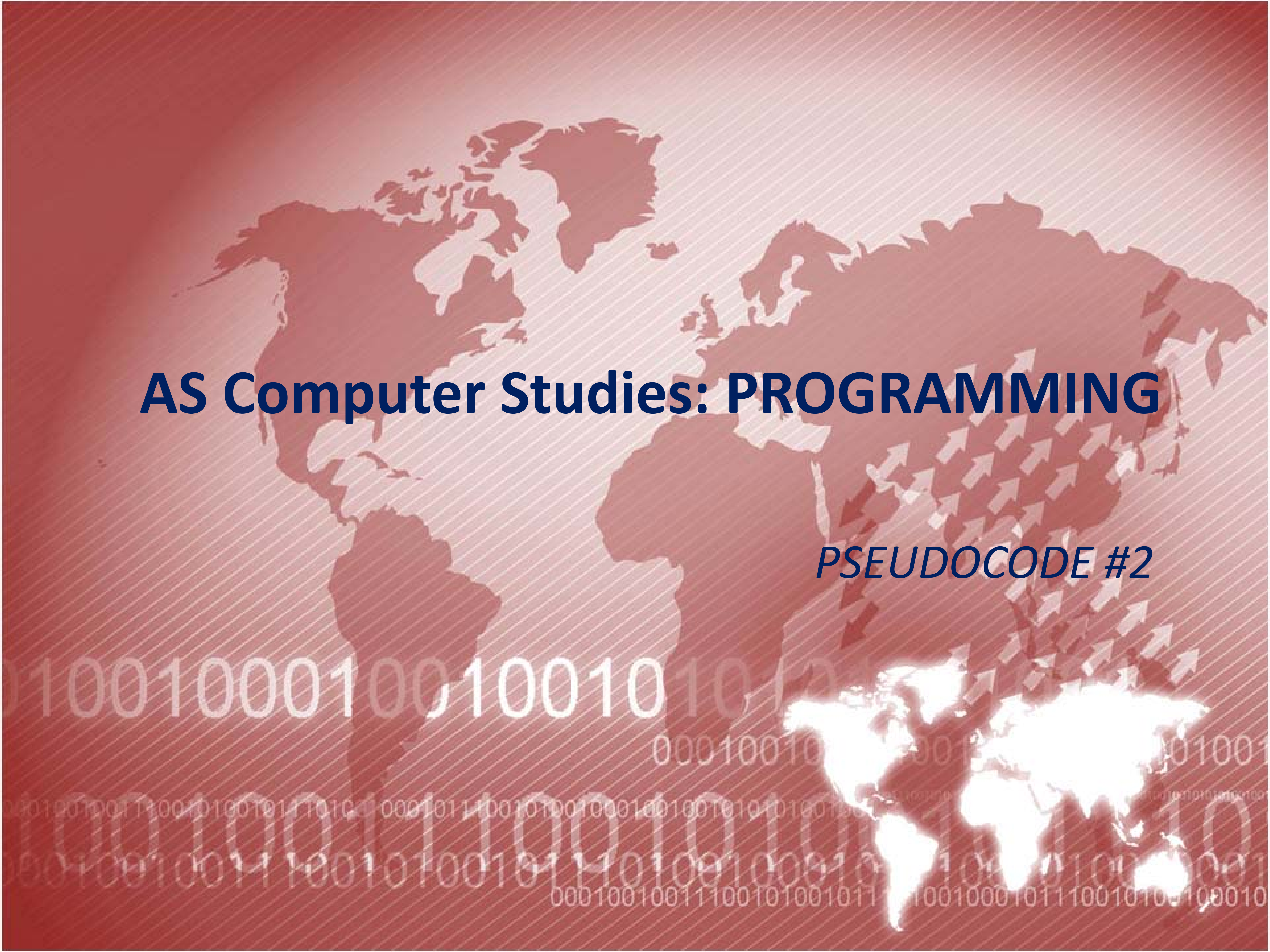


AS Computer Studies: PROGRAMMING

PSEUDOCODE #2

Learning Objectives

- Understand what **pseudocode** is and the difference from structured English.
- Describe the pseudocode **constructs** used in expressing an algorithm
- Use pseudocode to express an **algorithm**

Pseudocode

Pseudocode

A mixture of English and formatting to make the steps in an algorithm explicit

Algorithm to print all entries in a text file:

```
While (you haven't reached the end of file)
        Read current line from the file
        Print the line
        Go to the next line in the file
```

Following an Algorithm

Algorithm for preparing a Hollandaise sauce

```
If concerned about cholesterol
        Put butter substitute in a pot
Else
        Put butter in a pot
EndIf
Turn on burner
Put pot on the burner
While (NOT bubbling)
        Leave pot on the burner
Put other ingredients in the blender
Turn on blender
While (more in pot)
        Pour contents into lender in slow steam
Turn off blender
```

Developing an Algorithm

Two methodologies used to develop computer solutions to a problem

- Top-down design focuses on the **tasks** to be done
- Object-oriented design focuses on the **data** involved in the solution

But first, let's remember a way to express algorithms: pseudocode

Pseudocode

Pseudocode

A way of expressing algorithms that uses a mixture of *English phrases* and *indention* to make the steps in the solution explicit

There are no grammar rules in pseudocode

Pseudocode is not case sensitive

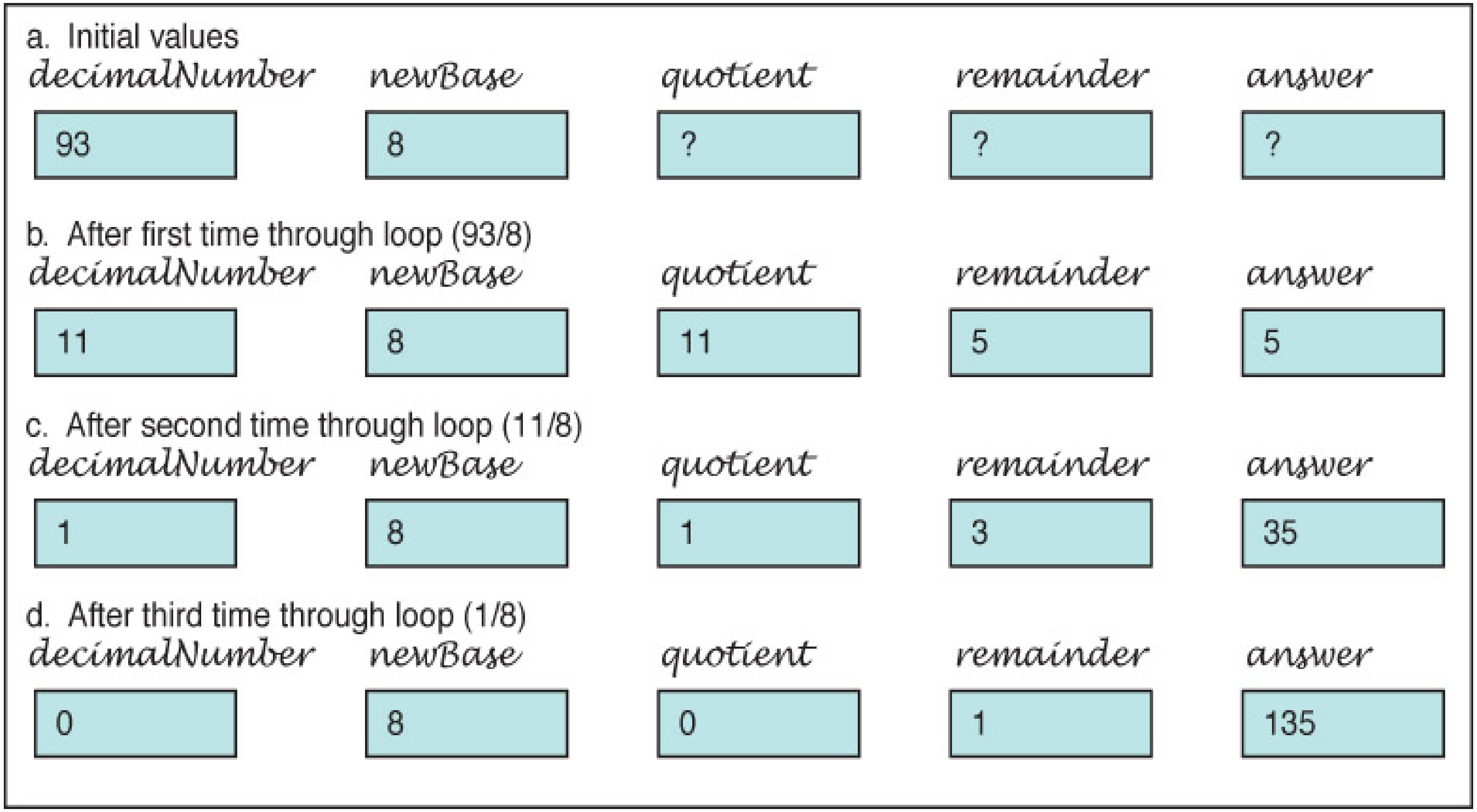
Following Pseudocode

```
While ( the quotient is not zero )
        Divide the decimal number by the new base
        Make the remainder the next digit to the left in the answer
        Replace the original decimal number with the quotient
```

What is 93 in base 8?

```
        93/8 gives 11 remainder 5
        11/6 gives 1 remainder 3
        1/ 8 gives 0 remainder 1
                        answer      1 3 5
```

Following Pseudocode



a. Initial values

decimalNumber	newBase	quotient	remainder	answer
93	8	?	?	?

b. After first time through loop (93/8)

decimalNumber	newBase	quotient	remainder	answer
11	8	11	5	5

c. After second time through loop (11/8)

decimalNumber	newBase	quotient	remainder	answer
1	8	1	3	35

d. After third time through loop (1/8)

decimalNumber	newBase	quotient	remainder	answer
0	8	0	1	135

Easier way to organise solution

Pseudocode for Complete Computer Solution

```
Write "Enter the new base"
Read newBase
Write "Enter the number to be converted"
Read decimalNumber
Set quotient to 1
While (quotient is not zero)
        Set quotient to decimalNumber DIV newBase
        Set remainder to decimalNumber REM newBase
        Make the remainder the next digit to the left in the answer
        Set decimalNumber to quotient
Write "The answer is "
Write answer
```

Pseudocode Functionality

Variables

Names of places to store values

quotient, decimalNumber, newBase

Assignment

Storing the value of an expression into a variable

```
Set quotient to 64
quotient <-- 64
quotient <-- 6 * 10 + 4
```

Pseudocode Functionality

Output

Printing a value on an output device

Write, Print

Input

Getting values from the outside word and storing them into variables

Get, Read

Pseudocode Functionality

Repetition

Repeating a series of statements

```
Set count to 1
While ( count < 10)
        Write "Enter an integer number"
        Read aNumber
        Write "You entered " + aNumber
        Set count to count + 1
```

How many values were read?

Pseudocode Functionality

Selection

Making a choice to execute or skip a statement (or group of statements)

```
Read number
If (number < 0)
    Write number + " is less than zero."
```

or

```
Write "Enter a positive number."
Read number
If (number < 0)
    Write number + " is less than zero."
    Write "You didn't follow instructions."
```

Pseudocode Functionality

Selection

Choose to execute one statement (or group of statements) or another statement (or group of statements)

```
If ( age < 12 )
        Write "Pay children's rate"
        Write "You get a free box of popcorn"
else If ( age < 65 )
        Write "Pay regular rate"
else
        Write "Pay senior citizens rate"
```

Pseudocode Example

```
Write "How many pairs of values are to be entered?"
Read numberOfPairs
Set numberRead to 0
While (numberRead < numberOfPairs)
        Write "Enter two values separated by a blank; press
  return"
        Read number1
        Read number2
        If (number1 < number2)
                Print number1 + " " + number2
        Else
                Print number2 + " " number1
        Increment numberRead
```

Walk Through

Data

3

55 70

2 1

33 33

numberOfPairs

Fill in values during each iteration

numberRead *number1* *number2*

What is the output?

Top-Down Design

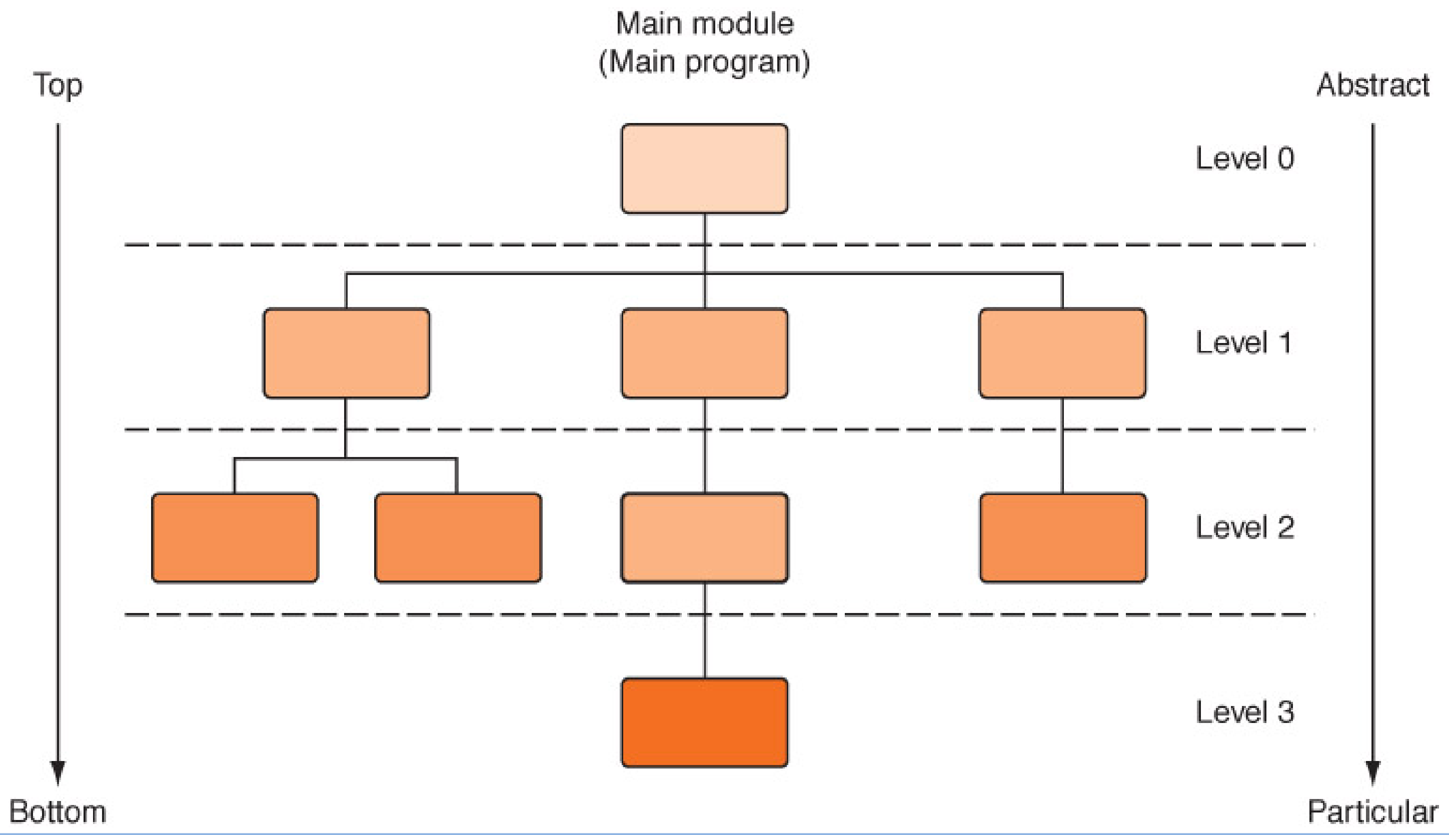
Top-Down Design

Problem-solving technique in which the problem is divided into sub-problems; the process is applied to each sub-problem.

Modules

Self-contained collection of steps, that solve a problem or sub-problem.

Top-Down Design



An example of top-down design

Process continues for as many levels as it takes to make every step concrete

Name of (sub)problem at one level becomes a module at next lower level

A General Example

Planning a large party

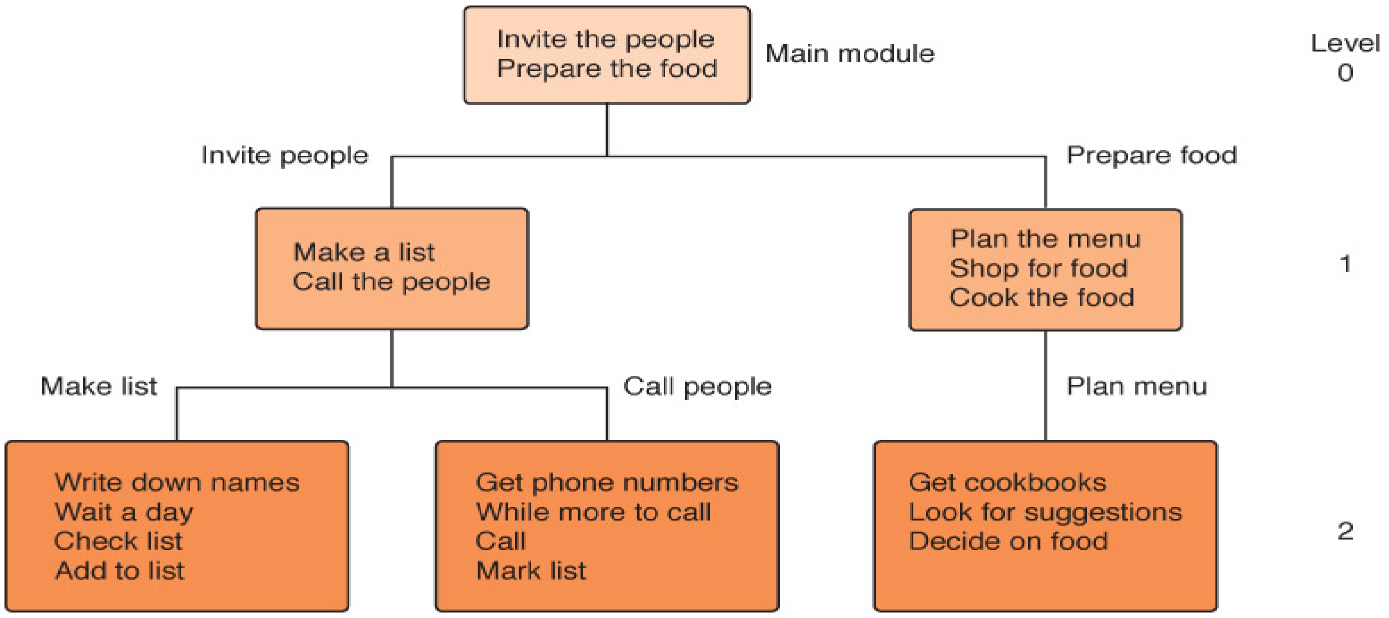


Figure 6.6 Subdividing the party planning

A Computer Example

Problem

Create a list that includes each person’s name, telephone number, and e-mail address

- This list should then be printed in alphabetical order
- The names to be included in the list are on scraps of paper and business cards

A Computer Example

Main — Level 0

```
Enter names and numbers into list
Put list into alphabetical order
Print list
```

Enter names and numbers into list — Level 1

```
While ( more names)
        Enter name
        Enter telephone number
        Enter email address
        Insert information into list
```

What is missing?

A Computer Example

<u>Enter names and numbers into list (revised)</u> Level 1

```
Set moreNames to true
While (moreNames)
        Prompt for and enter name
        Prompt for and enter telephone number
        Prompt for and enter email address
        Insert information into list
        Write "Enter a 1 to continue or a 0 to stop."
        Read response
        If (response = 0)
                Set moreNames to false
```

A Computer Example

Prompt for and enter name — Level 2

```
Write "Enter last name; press return."
Read lastName
Write "Enter first name; press return."
Read firstName
```

Prompt for and enter telephone number — Level 2

```
Write "Enter area code and 7-digit number; press return."
Read telephoneNumber
```

Prompt for and enter email address — Level 2

```
Write "Enter email address; press return."
Read emailAddress
```

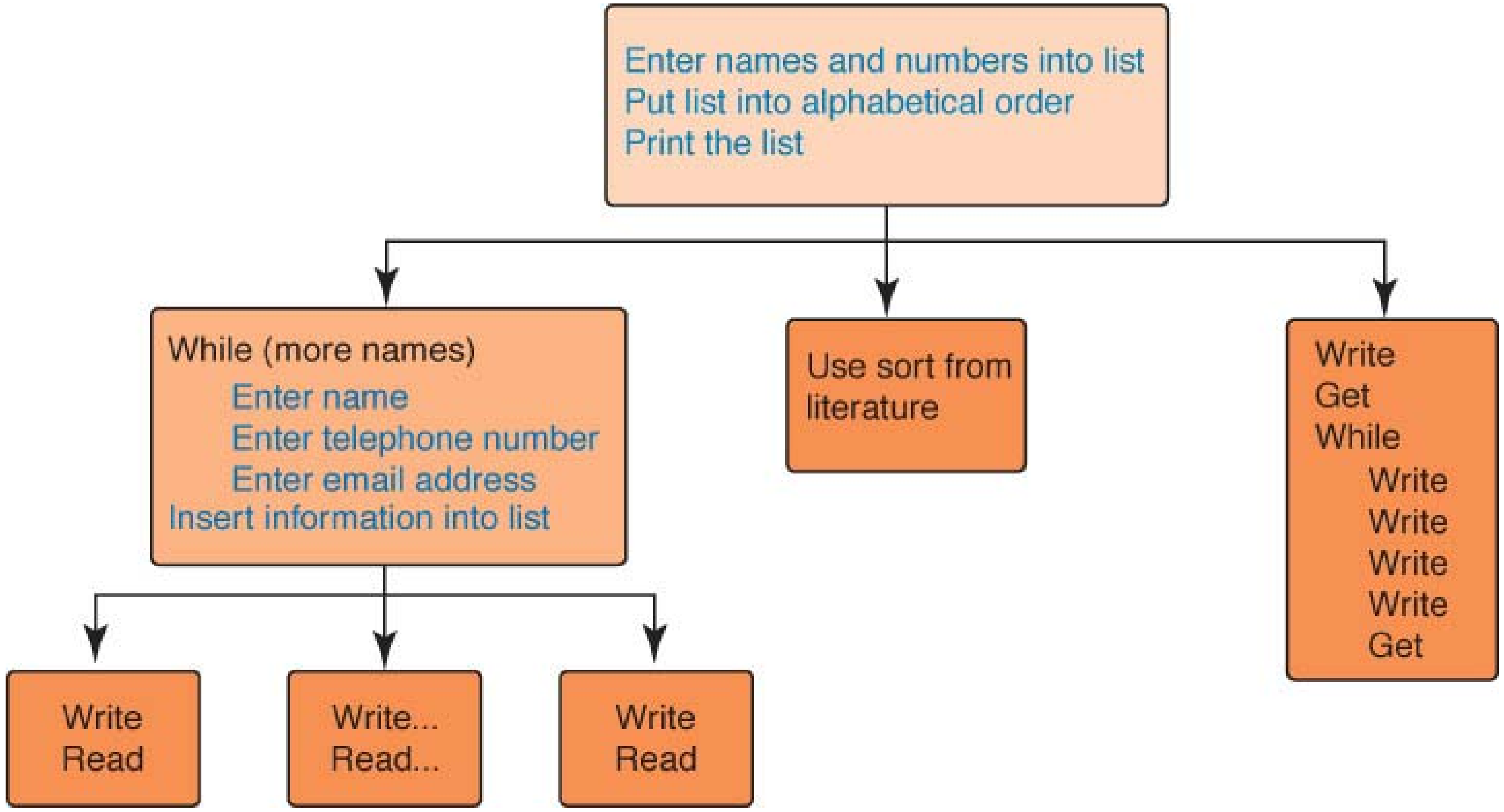
A Computer Example

Put list into alphabetical order

Print the list Level 1

```
Write "The list of names, telephone numbers, and email
        addresses follows:"
Get first item from the list
While (more items)
        Write item's firstName + " " + lastName
        Write item's telephoneNumber
        Write item's emailAddress
        Write a blank line
        Get next item from the list
```

A Computer Example



Note: Insert information is within the loop

Testing the Algorithm

Important distinction

Mathematics

We tests the *answer*

Programs

We test the *process*

Testing the Algorithm

Desk checking

Working through a design at a desk with a pencil and paper

Walk-through

Manual simulation of the design by team members, taking sample data values and simulating the design using the sample data

Inspection

One person (not the designer) reads the design (handed out in advance) line by line while the others point out errors