7 Programming

7.1 Algorithms

Understanding Algorithms

Computer science-type problems are solved using algorithms.

Once a problem has been specified, a series of steps or an algorithm can be created to produce a result.

Algorithms can be used to describe non-computer-type problems as shown in the travel example below.

Problem – Need to Travel to New York

1. contact travel agent and purchase ticket
2. collect travel documents
3. pack luggage
4. travel to airport
5. board correct plane
6. collect luggage
7. use taxi to leave airport

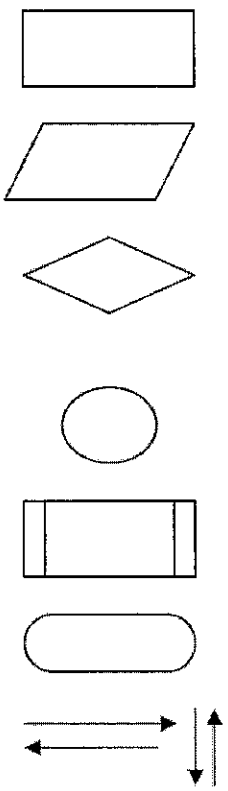
An **algorithm** is a step-by-step approach to solving a problem. It is written in pseudocode, which is independent of any particular programming language.

Flow charts or flow diagrams use a series of symbols and connecting lines to represent each step through an algorithm or system.

Obviously, the model could be refined to add further steps, such as choosing different routes, dates and price options.

Flow Charts

The following symbols are used in flow chart diagrams to show the steps through an algorithm or program.



Used to describe a process or operation in the program

Used to indicate any input or output of data in the program

In a decision diamond, there will be an input and two outputs where one represents true and the other false

Connecting circles are used to link parts of a flow chart that are drawn on more than one page

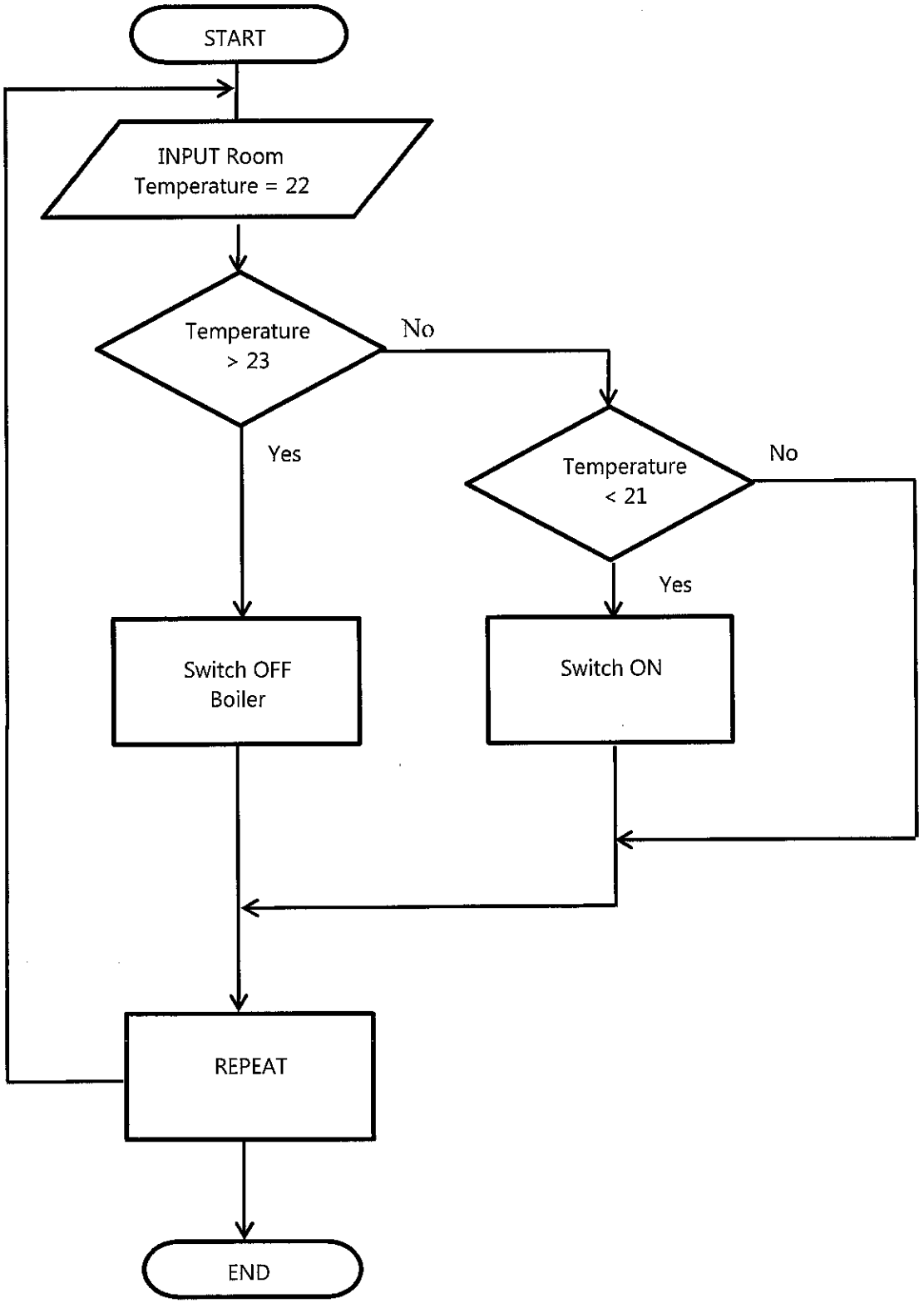
This symbol is used to indicate that a pre-defined subroutine has been used.

The lozenge-shaped symbol is used to indicate the start or end of a program

Flow lines are used with arrows to indicate the direction of flow through the program

 © ZigZag Education, 2013

Flow Chart Example – Use a flow chart to indicate how temperature is controlled in a central heating system.



In this case, the required temperature is selected as 22.

When the room temperature is higher than 22, the boiler is turned off.

When the room temperature is lower than 22, the boiler is turned on.

 © ZigZag Education, 2013

Pseudocode

An alternative method to using a flow chart is to use pseudocode to describe an algorithm.

> **Pseudocode** is an outline of code or a complete program that can easily be converted to the particular programming language being used.

There are no hard and fast rules for writing pseudocode, but algorithms written in pseudocode are easier to interpret when written using the following ideas:

- Capitalise the keywords: such as INPUT, OUPUT, IF, THEN, ELSE, END, REPEAT, UNTIL.
- Write one statement or action on each line.
- Use meaningful variable names where possible.
- Use indentations of statements to show structure clearly, for example:

```
IF a > 1
    b ← 5
ELSE
    b ← 1
END
```

Note – indentation is used in the average numbers example.

- Avoid programming language special features as the pseudocode should be language independent.

Pseudocode Average of N Numbers Algorithm

Example: Write an algorithm in pseudocode to find the total and average of N numbers, until a negative number is input.

The following steps take place:

1. Initially the variables average, count and total are zeroed.
2. The main calculations are carried out within the **REPEAT... UNTIL** N < 0 loop (in many programming languages this would be a DO... WHILE loop).
3. The first time through the loop, a number N is inputted and the variable count is used to keep track of the numbers inputted.
4. Next, the total and average are calculated and a new value of N is inputted.
5. When the value of N inputted is less than zero, the sequence stops and the count, total and average are outputted.

```
Average of N Numbers Algorithm
Average ← 0
Count ← 0
Total ← 0
REPEAT
        INPUT N
        Add 1 to Count
        Total ← Total + N
        Average ← Total / Count
UNTIL N < 0
OUTPUT   Count,   Total,   Average
```

Meaningful variable names are used to aid the interpretation of the function.

> Note that the left arrow symbol (←) is normally used to make an assignment in pseudocode, rather than the equals (=) symbol.
> The left arrow can be inserted into most word processors using shortcut **ALT 27** (hold down the Alt key and press 27 on the number pad).

 © ZigZag Education, 2013