# Chapter 2
# Flowchart

## 2.1 What is a Flowchart

A program may be understandable by only developers and engineers. In contrast, a flowchart is a diagrammatic method that enables anyone to understand a program or calculation procedure. In this book, we will use flowcharts to explain many calculation procedures.

In a flowchart, each step is expressed by a box and the flow is shown by arrows between boxes. For example, Fig. 2.1 is a flowchart of the Java program shown in Fig. 1.3. In the flowchart, “Integer $a = 5$” corresponds to “int $a = 5$” in the program, “Integer $b = 3$” corresponds to “int $b = 3$” in the program, and “Output the sum of $a$ and $b$” corresponds to “System.out. println($a + b$)” in the program. Since the steps in the flowchart are easily understandable, anyone reading it can grasp the program’s functionality.

Initially used in the field of programming, flowchart is now widely used in various fields. For example, Fig. 2.2 is a flowchart illustrating the procedure of washing hands. First, wet your hands with water. Next, lather them with soap. Then, rinse off the soap with water. Afterward, check if your hands are clean. If yes, dry your hands with a towel and stop. Otherwise, repeat the above process.

A flowchart containing business details is useful in discussions with business partners and can often be understood more easily than a spoken explanation. It is important that the statements in a flowchart should be both concise and clear in meaning, as shown in Fig. 2.2.

Typical “boxes” in a flowchart include terminals, input/output, decision, and processes [1]. Figure 2.3 shows the symbols of these boxes. It should be noted that the symbols are predetermined and constant. “Start” and “end” are called terminals and should be oval. It is highly recommended to use them, because they serve as clear entry and exit points for the flowchart. Input and output are optional and should be parallelograms. Input refers to information going into the computer, such as data from the user. Output shows the result of a calculation or a message on the

W. Weng, *A Beginner’s Guide to Informatics and Artificial Intelligence*,
https://doi.org/10.1007/978-981-97-1477-3_2

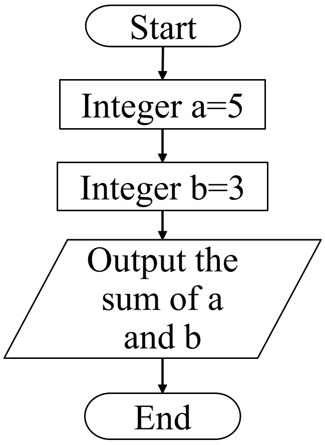

**Fig. 2.1** A flowchart for the program shown in Fig. 1.3

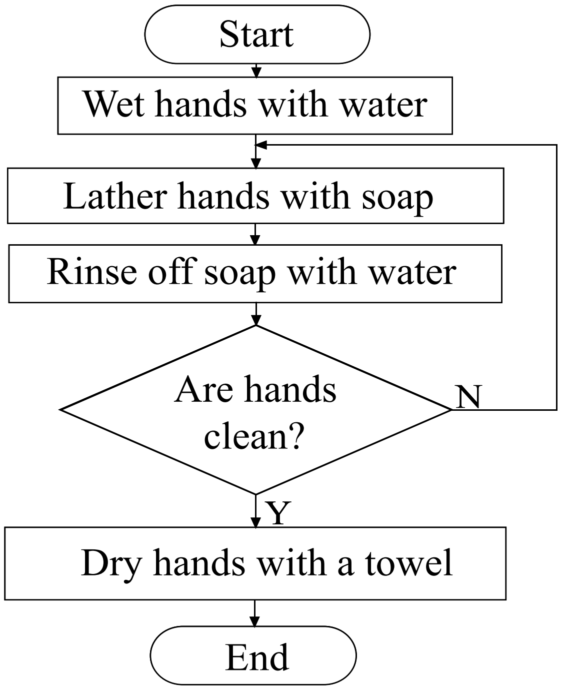

**Fig. 2.2** A flowchart of washing hands

display. Decision is optional but is often used. A decision box should be a diamond and should have two output arrows, which typically represent the answers “yes” and “no”. Process boxes are required. A process box should be a rectangular in which details of a step are described.

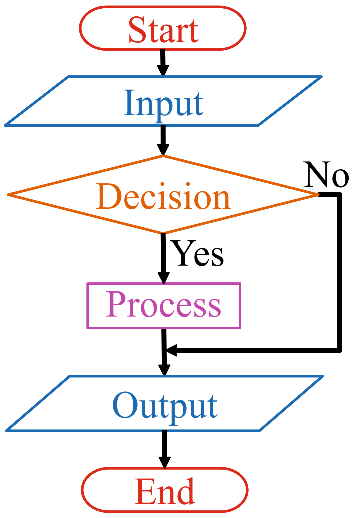

**Fig. 2.3** Typical boxes in a flowchart

## 2.2 How to Create a Flowchart

The general approach to create a flowchart is explained using Fig. 2.4, a flowchart of calculating the sum of integers 1 to $n$.

First, determine whether input and output are necessary. To get the sum of $n$ integers, $n$ should be specified by the user, so input is necessary. The sum should be displayed, so output is also necessary. As a result, create the boxes "input $n$" and "output $s$ (sum)". Then initialize variables, i.e. set the initial value of every variable used in a flowchart. The output "$s$" is a variable (the reason why $s$ is a variable will be given later) and hence should be initialized. "$s = 0$" is its initialization, which sets the initial value of $s$ to 0. After initialization, create process boxes to detail the calculation procedure.

To calculate "$1 + 2 + 3 + \ldots$", a variable $i$ should be created to represent each integer such as 1, 2, and 3. "$i = 1$" sets the initial value of $i$ to 1. After initializing $i$, the calculation procedure starts.

The meaning of the symbol "=" in a flowchart is to assign the value on the right side to the variable on the left side. If the right side is a number, the number is assigned directly to the variable on the left side, as exemplified in the steps "$s = 0$" and "$i = 1$". If the right side is a formula, then calculate the formula first and then assign the result to the variable on the left side, as exemplified in the step "$s = s + i$". This step calculates "$s + i$" first and then assigns the result to $s$. Since the initial value of $s$ is 0 and the initial value of $i$ is 1, this step calculates "$0 + 1$" first and then assigns the result 1 to $s$. In other words, after executing this step, $s$ changes from 0 to 1. Therefore, $s$ is a variable that stores the sum of integers 1 to $i$.

If $n > 1$, the next step will be adding 2 to $s$. Since the current value of $i$ is 1, it should be increased by 1 so that $i$ can represent 2. This is achieved by the step "$i$

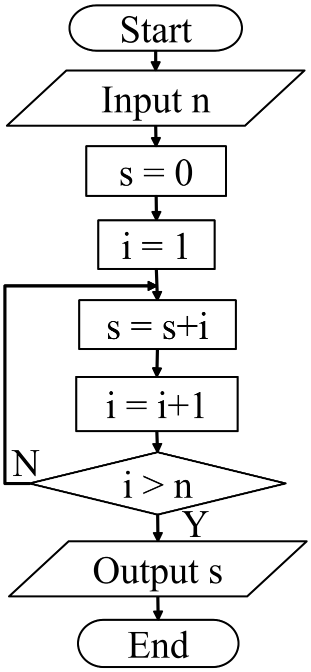

**Fig. 2.4** A flowchart of calculating the sum of integers 1 to *n*

$= i + 1$". Then create a decision box to check if the current value of $i$ has reached $n$. Supposing that $n$ is 10, then the current value of $i$, 2, has not reached $n$, so the procedure exits the decision box from the arrow "no" and returns to "$s = s + i$". As a result, "$s = s + i$" will be executed again. This time, the step adds 2 to $s$ and updates $s$ with the result 3. Hence, after executing this step, $s$ becomes the sum of integers 1 to 2.

The process of increasing $i$ by 1 and adding it to $s$ will be repeated until $i$ becomes greater than $n$. When $i$ becomes $n + 1$, the procedure will exit the decision box from "yes" and output $s$.

Statements in a flowchart can be conveyed through textual descriptions as shown in Fig. 2.2, or through the use of variables as illustrated in Fig. 2.4. When conveying numerical information, variables are commonly employed due to their ability to provide concise and precise representation compared to words. A flowchart can incorporate a blend of textual descriptions and variables, as exemplified in Fig. 2.5, a flowchart of taking fitness training ten times.

## 2.3 Notes on Flowcharting

Flowcharts created by beginners often have problems. Here are some notes you should pay attention to when creating flowcharts.

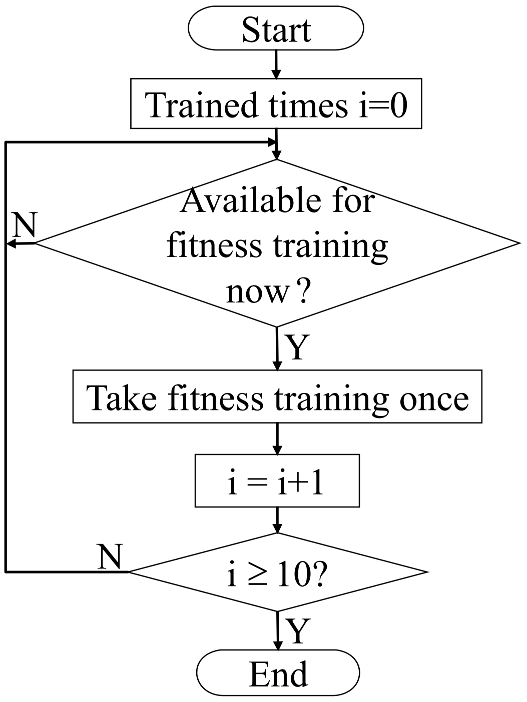
Start
Trained times i=0
Available for fitness training now ?
N
Y
Take fitness training once
i = i+1
i ≥ 10?
N
Y
End

**Fig. 2.5** A flowchart of taking fitness training 10 times

First, a decision box typically presents a yes/no question and has two output arrows corresponding to the answers to that question. Hence, create a question that can be answered by yes or no for each decision box. If a decision cannot be made with a simple yes/no answer, consider using multiple yes/no questions, as exemplified in Fig. 2.6. A decision box should have one more output arrow than the other boxes, so make sure to include the second output arrow. It is easy for beginners to believe that the box has been completed after drawing only one output arrow. In addition, make sure to label the two arrows “yes” and “no”. If you revise the question in a decision box or the steps connected to it, check again if the labels “yes” and “no” are still correct after the update. It often happens that the labels become reversed after updating some boxes.

Second, do not write more than one question in a decision box. If you have multiple questions, create multiple decision boxes and write one question in each of them. For example, if you want to express three cases: $i < 10$, $i = 10$, and $i > 10$, avoid creating a flowchart like the one depicted in Fig. 2.6a; instead, create it in the way shown in Fig. 2.6b.

Third, boxes other than decision boxes should typically have a single output from the bottom. Having two or more outputs can lead to confusion, as it implies simultaneous execution of multiple steps, which contrasts with the purpose of a flowchart—to illustrate the sequential order of execution.

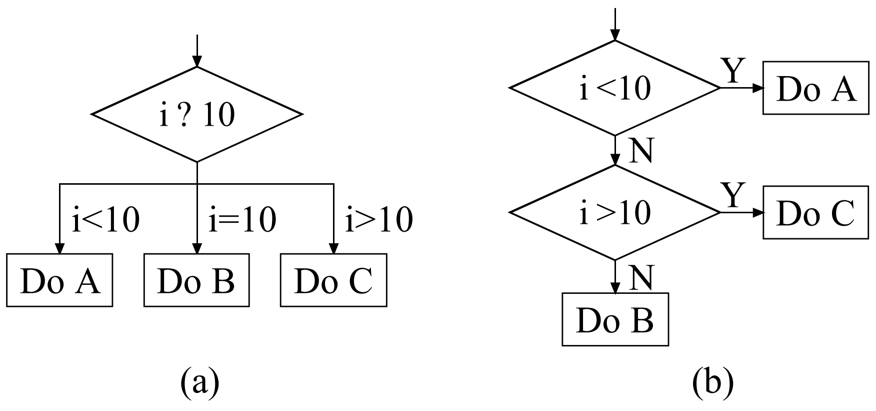

**Fig. 2.6** An example of using multiple decision boxes

Last, it is generally preferred that arrows enter boxes from the top, rather than from the left or right side. This creates a visual hierarchy that aligns with the top-to-bottom reading habit in many cultures, improving clarity and understanding. Figure 2.7 shows two examples in which arrows enter boxes from the top and the right side, respectively. While it is true that many drawing software programs allow arrows to enter boxes from the left or right side, this flexibility is often provided to accommodate various diagramming needs. It is important to ensure that the flowchart remains clear and understandable.

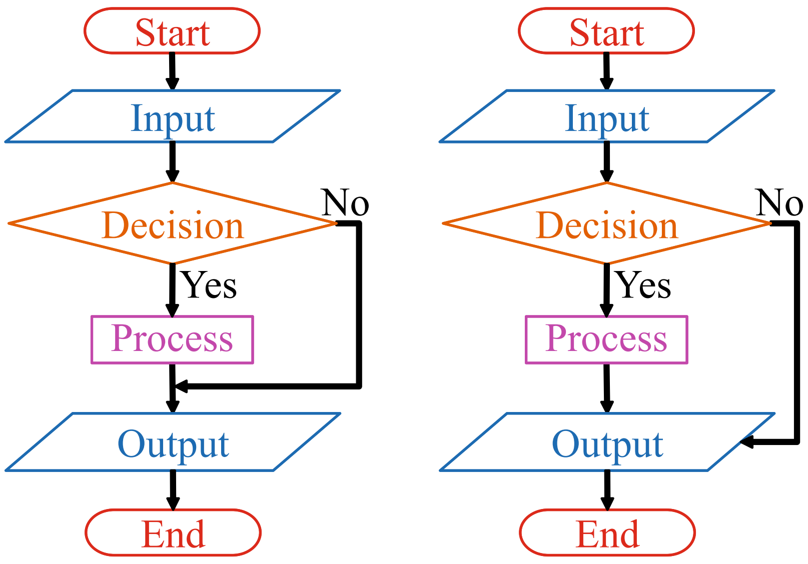

**Fig. 2.7** Examples of arrows entering boxes

## 2.4 Clarify the Meaning of Each Statement

A flowchart should be easily understandable and free from ambiguity. Each statement should have a clear and singular meaning.

For example, using a single word like “jump” in a process box can lead to confusion. Is it a forward jump or upward? With one foot or two? For how long or how many times? In contrast, a statement such as “jump up once with both feet” will clear up most of the confusion. As another example, using “cook” as a process box is insufficient because it lacks specificity. It would be much clearer to provide details such as “bake at 230° for 10 s”.

Similarly, ensure clarity in decision criteria. Avoid creating conditions that are hard to judge. For example, a question like “Is it late?” is insufficient because the criteria for “late” are unclear. In contrast, a condition such as “Is it later than 9 p.m.?” would be much better.

In short, a good flowchart should enable the reader to understand and execute its instructions with clarity and ease.

### Exercises

1. Create a flowchart detailing a jumping exercise procedure where you should jump up either 15 times or for 20 s, whichever comes first.

**• Guidance**

Some learners may create a flowchart like the one shown in Fig. 2.8. The figure lacks details, but if you create a similar flowchart, there is room for improvement. Figure 2.8 has many problems, such as having two output arrows exiting a box that is not a decision box. We discussed this problem in Sect. 2.3. Non-decision boxes should have only one output because multiple outputs cannot be executed simultaneously. Therefore, it is better to use a single output and evaluate the two conditions sequentially.

In addition, the steps required before you can tell the answers to the two questions are not included in Fig. 2.8. If you do not count, you will not know if 15 times have been reached. The procedure of counting should be included in the flowchart. You may refer to the way of writing in Fig. 2.5. You may use $i$ as the counter and after each jump, increase $i$ by 1. By comparing $i$ with 15, you can know if 15 times have been reached. Since $i$ is a variable, you should initialize it first. Similarly, you should create a step for setting a timer or an alarm so that you can know if 20 s are reached. If you write all these details, you will get the full score.

The exercises in this and subsequent chapters serve the dual purpose of assessing your understanding and enhancing problem-solving abilities. It is crucial to develop the skills to apply classroom knowledge to unfamiliar problems. Although it

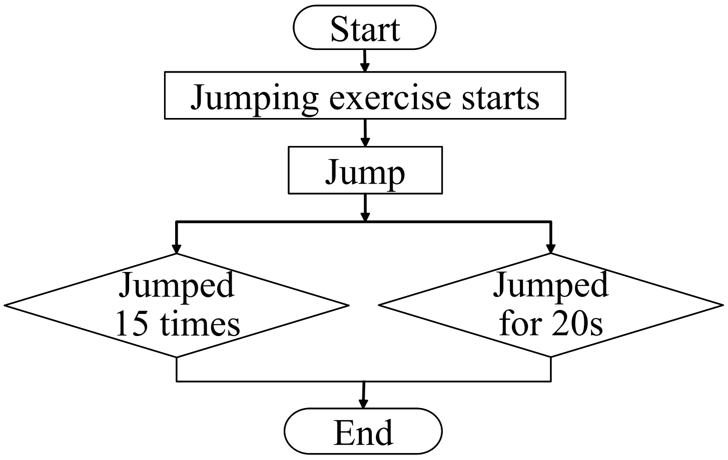

**Fig. 2.8** A flowchart created by some learners

may be challenging initially, dedicated effort often leads to greater-than-expected achievements.

## Reference

1. Andersen B, Fagerhaug T, Henriksen B, Onsøyen LE (2008) Mapping work processes, 2nd edn. ASQ Quality Press, Milwaukee