ICT Level 2 – Program Design – Flow Charts

1 of 14 – Welcome

Welcome to this session on program design: flow charts.

By the end of this session, you will:

* Understand what flow charts are
* Know and understand flow chart symbols for programming
* Understand how flow charts are used to design computer programs

2 of 14 – Introduction to program design: flow charts

**What is meant by program design?**

Every program that is installed onto a computer system first has to be designed and tested, chiefly to make sure that it is safe to use, but also to ensure that the program is providing the right service or task (which is, after all, the point of adding new software to a system).

When a program has been designed, a mock test can be carried out to see whether the individual parts of the program are working properly.

While there are a number of approaches for designing computer programs, one of the more popular methods is using a flow chart.

3 of 14 – An introduction to flow charts

Flow charts are a common way of mapping out certain events and processes.

By using a flow chart as part of program design, programmers are able to **plot each part of a program onto**

**the chart by assigning it its own space**. They can then track the activity of each individual component by connecting it to another part, and so on, until the entire program layout is mapped onto the chart structure.

By plotting the individual elements of a program like this, programmers can then **see all of the individual processes involved**, such as: where the program begins, when and where the decisions are being made, how operations are processed, and what the inputs and outputs might be.

4 of 14 – Flow chart symbols

To make the flow charts in computer programming a little easier to read, programmers use different boxes to mark different functions.

This means that an instruction written inside a rectangular box will mean something entirely different to that same instruction written inside a square box.

This helps to make it even clearer what each individual component of the program should be doing at any given time, because its box will describe its purpose within the entire system.

The different types available are **process**, **terminator**, **decision box**, **data (input/output)**, and **connector**.

We will look at these in more detail in the following sections.

5 of 14 – Flow chart symbols examples

**Process**

The box for Process in a computer programming flow chart is just a simple **rectangle**.

**Remember:** flow charts are particularly useful for illustrating where operations are being processed within a program design.

Process boxes are used to **highlight a calculation or process** that is being performed by the program.

Another way to think of these process boxes is that they contain an instruction or a command, which will then be completed or finalised by another part of the program.

**Terminator**

The box symbol for Terminator is a **rectangle** again, but this time with **rounded edges** – creating an obvious distinction between terminators and processes.

The terminator box is sometimes referred to as a **start/stop box**, as this is what the box indicates. They are used for the start and end points within a program.

**Remember:** it is possible to have more than one start and end point in a program, and so it is likely that there will be more than just two of these features in any program design flow charts.

**Decision Box**

The Decision Box and data boxes in these flow charts can sometimes be a little confusing, as certain designs can make these two shapes look very similar.

However, as long as the Decision Box resembles a **diamond** – then a user can assume it indicates a decision instruction.

In computer programming, these decisions are when the program is **testing where to move next**, and also **where to direct the next coded instruction**. Another way to think of these is as **yes or no moves**; the program decides whether a certain decision was made by the user (did they or did they not use a certain key, for example), and then the program moves forward in the appropriate direction.

**Data (Input/Output)**

Data (Input/Output) can be abbreviated to **Data (I/O)** and this can be found on a flow chart by looking for the **parallelogram shaped boxes**.

These boxes are used to show the points at which the program either **takes in** or **distributes out** data.

The inputs might be something that have to be added by the user – such as writing in additional details, for example – while the outputs might be things that are printed or somehow illustrated on the computer’s screen.

**Connectors**

Connectors do not have a box on the flow chart but are instead marked by an **arrow**.

These arrows – or rather, connectors – are sometimes known as **flow lines**. Both of these names reveal their function on the flow chart, as they refer to the image used to connect parts together and indicate the possible **routes** running through the entire chart.

6 of 14 – How these pieces fit together

Below is an example of how flow chart elements may connect together:

1. The flow chart starts using the rounded rectangular shape
2. A connector arrow is used to connect the start box to the next shape, a data input
3. A data input is next in the flow chart using a paralleogram shape
4. A connector arrow is used to connect the data input box to the next shape, a decision
5. A decision is next in the flow chart using a diamond
6. Two connector arrows are used to connect the decision box to other flow chart elements, one connector is labelled yes, and one is labelled no

This pattern of data input and output and decision making will continue for the length of the program. A good example of this is to consider the games played on a computer, and how a series of moves and decisions are made until eventually the user reaches the end (using a rounded rectangular shape).

7 of 14 – Advantages to using a flow chart

Using a flow chart to plot out the structure and design of a computer program means that programmers get **a good overview** of which component is performing each task.

It is also worth noting that not only are flow charts useful from the perspective of organising a program, but they can also **make it easier to debug a program** too. Problems can be isolated and analysed more easily, and this can also save programmers time (and potentially money) in the long run.

Flow charts are also useful for program maintenance. Having an outline of the program from start to finish can be useful for sectioning off different components that might need to be upgraded or altered, and this is another way that flow charts make for time efficient program design.

8 of 14 – Disadvantages to using a flow chart

While many computer programmers rely on flow charts, there are some disadvantages to using them that should be considered.

They are not necessarily the easiest way to plan a program. While some program designs work well on a flow chart, some are simply **too complicated**, meaning the flow chart also becomes complicated and can become a clumsy explanation for what the program is doing (if there are too many connectors going in too many different directions, for example).

Also, while they can save time in many ways, they can cost time too. Consider the **time** it might take to entirely redraw a flow chart, all for the sake of one small design change – plus, if the program is an intricate one, this will be even more time consuming.

So while they are useful for designing programs, it does not mean all programs are suited to them.

9 of 14 – Question 1

What does a rounded rectangular shape mean when used on a flow chart?

1. That is a process box used to indicate a process or calculation
2. That is a terminator used to indicate a start or finish
3. That is a data input or output
4. That is a connector that ties together the program

The correct answer is B, this is a terminator used to indicate a start or finish.

10 of 14 – Question 2

Categorise the advantages and disadvantages of flow charts below according to if they are an **advantage** or a **disadvantage**.

Helps with maintenance

Not always suitable

Can take too much time

Easier to debug

Can become clumsy

Provides a good overview

The correct answers are:

Helps with maintenance, easier to debug and provides a good overview are **advantages** of flow charts.

Not always suitable, can take too much time and can become clumsy are **disadvantages** of flow charts.

11 of 14 – Question 3

What does a diamond shape mean when used on a flow chart?

1. It indicates a decision is being made
2. It indicates an end point in the program
3. It connects one part of the program to another
4. It indicates data is being input or output

The correct answer is A, it indicates a decision is being made.

12 of 14 – Question 4

Using the following choice of words; **overview**, **complicated**, **popular**, **connected** and **time-consuming**, fill in the blanks for the paragraph below:

While there are many different ways to design a computer program, flow charts are particularly **blank**. This is because they provide a detailed **blank** of the individual parts of a program, while also explaining how these different parts are **blank**. Through using different shapes on the flow chart, programmers can even highlight the type of function in each step as well, such as data input or a decision being made. However, while flow charts are very useful, they may not always be appropriate for more **blank** designs. If there are many processes within the program, for example, the chart can become clumsy and **blank**.

The correct paragraph should read:

While there are many different ways to design a computer program, flow charts are particularly **popular**. This is because they provide a detailed **overview** of the individual parts of a program, while also explaining how these different parts are **connected**. Through using different shapes on the flow chart, programmers can even highlight the type of function in each step as well, such as data input or a decision being made. However, while flow charts are very useful, they may not always be appropriate for more **complicated** designs. If there are many processes within the program, for example, the chart can become clumsy and **time-consuming**.

13 of 14 – Question 5

On a flow chart structure, what does an arrow mean?

1. It is a flow line that marks problem areas within a program
2. It is used to show where data is being input into the program
3. It is a connector that is used to explain the flow of the program
4. These lines are only ever used to indicate yes or no after a decision box

The correct answer is C, it is a connector that is used to explain the flow of the diagram.

14 of 14 – End

Well done. You have completed this session on program design: flow charts.

In this session we have covered:

* What flow charts are
* The flow chart symbols for programming
* How flow charts are used to design computer programs

If you have any questions about any of these topics, make a note and speak to your tutor for more help.