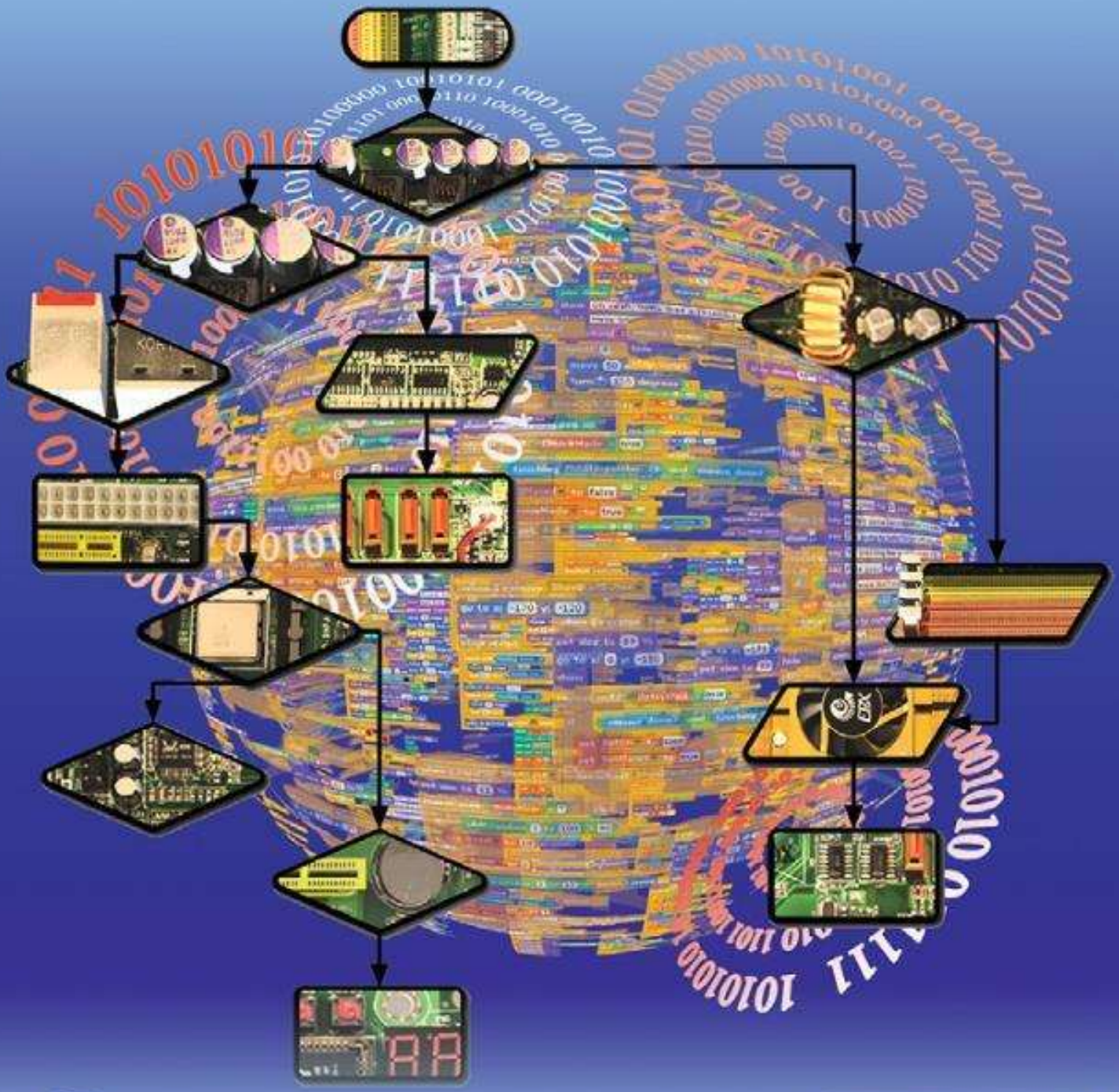
IT is gr8! @ Grade 10



Study OPPORTUNITIES

CAPS edition

IT is gr8! @ Grade 10

Ulza Wassermann Chris Noomé

Edward Gentle Keith Gibson Pam Macmillan Malie Zeeman

Published by Study Opportunities
PO Box 52654, Dorandia, 0188 • Tel: (012) 565-6469 • Fax: (012) 546-5313
E-mail: study.opp@mweb.co.za • www.studyopportunities.co.za
November 2011 ISBN: 978-1-920535-07-0
DTP, layout and design by Ronelle Oosthuizen • Cover design by Suzanne Jacobs

This book is dedicated to the memory of our friend and colleague Annette Bezuidenhout who lost her brave fight against cancer in 2009. She inspired many to believe that IT is great!

Contents

Introduction 1

Section 1: Algorithm Design and Programming 7

Module 1.1 Thinking for computers 8
Module 1.2 Introduction to graphical programming 25
Module 1.3 Input, processing, output 43
Module 1.4 Decision making 59
Module 1.5 Interaction between sprites 75
Module 1.6 Lists 83
Module 1.7 Prevent and find errors 90
Module 1.8 Projects 95

Section 2: Systems Technologies 105

Module 2.1 Basic concepts of computing 106
Module 2.2 Basic concepts of hardware 118
Module 2.3 Basic concepts of system software 136
Module 2.4 Computer management 143
Module 2.5 Improve your browser 154

Section 3: Communication and Internet Technologies 159

Module 3.1 Networks 160
Module 3.2 Electronic communication 172
Module 3.3 Internet and WWW 182
Module 3.4 Web page technology 196

Section 4: Data and Information Management 203

Module 4.1 Data as the computer sees it 204
Module 4.2 File management 213

Section 5: Social Implications 225

Module 5.1 Social, ethical and legal issues surrounding ICTs 226
Module 5.2 Green, health and global e-communication issues 231
Module 5.3 Safe Internet and e-mail use 236

Appendix: Connectors on a computer 247

Table of Contents

INTRODUCTION

Introduction 1

SECTION 1
Algorithm Design and Programming

Module 1.1 Thinking for computers ... 8
Computers and 'thinking' 8
Create a solution for a problem 9
Algorithms for simple mathematical problems 15
Algorithms with decisions 19
Algorithms with repetition 21

Module 1.2 Introduction to graphical programming 25
Gather your resources 25
Get to know the Scratch environment 26
Teach a sprite to perform basic tasks 29
Teach a sprite to repeat actions 32
Create animation 35

Module 1.3 Input, processing, output 43
Introduction 43
Teach a sprite to 'remember' by using variables 43
Teach a sprite to do calculations with numbers 47
Teach a sprite to handle strings 54

Module 1.4 Decision making 59
Introduction 59
Decision-making in Scratch 59
Decide when to terminate the loop 71
Decide when to enter the loop 73

Module 1.5 Interaction between sprites 75
Broadcasting and receiving messages 75
Two sprites communicate by sharing data 79

Module 1.6 Lists 83
Representation of a list 83
Create a new list 83
Program blocks for lists 84
Store items in a list 84

Module 1.7 Prevent and find errors 90
Data validation 90
Debugging your program 92

Module 1.8 Projects 95
Special effects 95
The importance of planning 95
The final touches 96
Follow the design of a coin toss game 97
Flags and interactive elements 102

SECTION 2
Systems Technologies

Module 2.1 Basic concepts of computing 106
What is a computer? 106
An expanded model of a computer 107
The role of software 109
Types of computers 112
ICT systems 114

Module 2.2 Basic concepts of hardware 118
Inside the case of a computer 118
Connecting to the computer 120
Peripherals 122
Smartphones versus computers 133

Module 2.3 Basic concepts of system software 136
Introduction 136
Operating system 136
Utility programs 139
Device drivers 139

Module 2.4 Computer management 143
Introduction 143
Initial maintenance 143
Ongoing maintenance 146
Adding and removing software and hardware 148
Other housekeeping tasks 149
Module 2.5 Improve your browser 154
Modular software 154
Browser plug-ins 154

SECTION 3
Communication and Internet Technologies

Module 3.1 Networks 160
What is a network? 160
Reasons for having a network 160
Disadvantages of having a network 162
Components of a network 162
Types of networks 166
Client-server vs. peer-to-peer LAN 168
Access control when logging into a network 170

Module 3.2 Electronic communication 172
Applications to facilitate e-communication 172
E-mail 175
Practical use of e-mail 177

Module 3.3 Internet and WWW 182
What is the Internet? 182
What is needed to connect to the Internet? 182
The World Wide Web 184

Module 3.4 Web page technology 196
Introduction 196
What is a web page? 196
HTML 197
Basic formatting tags 198
Characteristics of a well designed web page 199

SECTION 4
Data and Information Management

Module 4.1 Data as the computer sees it 204
Number systems 204
Data as the computer sees it 204
Data storage 208

Module 4.2 File management 213
Introduction 213
The file manager 213
Files – the basic unit of storage 216
The need to organise 220

SECTION 5
Social Implications

Module 5.1 Social, ethical and legal issues surrounding 226
ICTs 226
Introduction 226
EULAs, copyright and piracy 226

Module 5.2 Green, health and global e-communication issues 231
Ergonomics and health issues 231
The need for green computing 232
The impact of global e–communication 234

Module 5.3 Safe Internet and e-mail use 236
Introduction 236
Viruses 236
Computer worms 238
Trojans 238
Spyware 238
Adware 239
Spam 239
Hoaxes and scams 240
Phishing 241
Pharming 241
E-mail spoofing 241
Guidelines using the Internet and e-mail with courtesy and safety 242

APPENDIX

Connectors on a computer 247

Introduction

Information Technology (IT) sounds so dry and boring. And, like most things, you can approach it in a dry and boring fashion...

But what IT really is goes beyond our wildest dreams and imagination. IT is touching glass with your fingertips and watching images move in response, or watching the WWW scroll by beneath your hand. IT is moving around in front of your game console and watching your game character do exactly what you do as you do it. IT is also being able to search through millions of data records in milliseconds and find that valuable piece that you need quicker than any human could possibly find it. IT is doing calculations in minutes that would take teams of hundreds of skilled mathematicians a decade to do. IT creates worlds like Pandora (in the movie Avatar) and lets you view streets around our own world using Google Street View. IT allows you to listen to music wherever you go on your MP3 player. IT stops the brakes on your car from locking, and so helps to prevent a possibly fatal skid when you have to stop suddenly and quickly.

Information and Communication Technology (ICT) includes all the technologies that allow fast, efficient communication using spoken and written words, as well as the broadcasting of pictures and video around the world in record-breaking time.

IT is gr8! – and, as anyone knows,
it takes a lot of studying and skill and practice
to learn how to create something great!

The aim of this year's study is twofold:

- To help you understand a bit more about computers – what they consist of, how they work and what they can do and are used for.
- To teach you to solve problems in ways that will allow you to create computer programs to execute al sorts of wonderful tasks.

Introducing... ROTFL Cat

ROTFL Cat says:

Hi, I'm ROTFL Cat (LOL Cat's bigger brother). I'm here to give you handy tips and to highlight important concepts and definitions for you to pay attention to.

It's a good idea for you to pay extra attention whenever I appear. I've been around ICT for a while and so the tips I give you can be very handy!

Before we get started, you have to learn something very important – and you have to be able to apply it to everything you do and learn in the next three years of studying IT.

You have to start to analyse and solve problems in such a way that you will be able to give instructions to a computer to solve similar problems.

- When we provide data to the computer, we call it **input**.
- When the computer provides us with information, we call it **output**.
- All the steps performed to solve a problem is called **processing**.

The way to do this is to learn to

- determine which facts you have and what new information you want to obtain when the problem has been solved
- break the solution down into small steps (one action per step)
- arrange the steps in the right order, one after another
- make decisions in terms of questions that have only 'Yes' or 'No' as an answer

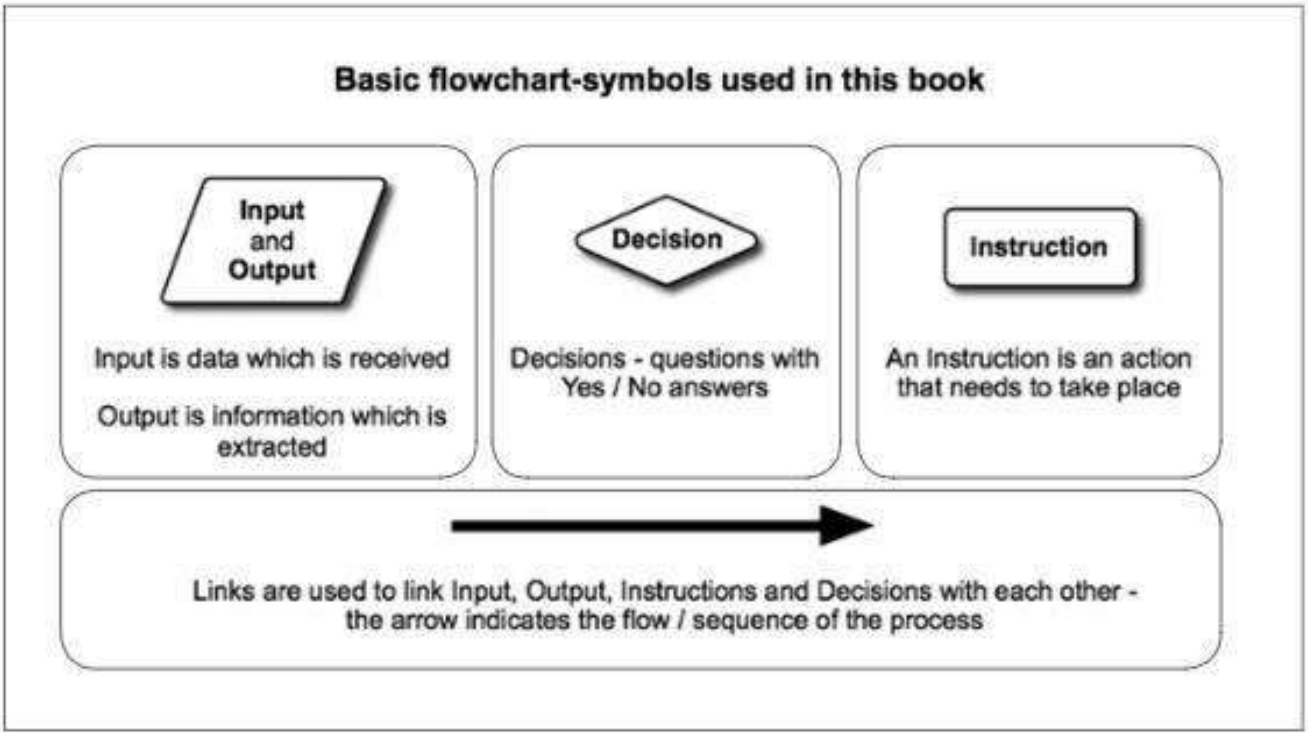
You can apply this technique to any problem you come across. In this way, you will train yourself to think in such a way so that you can teach a computer – or "to '**think**' like a computer", as we will be phrasing it.

Our tool for learning to represent the way that a computer 'thinks' and works will be a **flowchart** or 'decision diagram'. This is a graphical, visual representation of a process, showing tasks that are performed, decisions that are made and the order (sequence) in which these happen.

The flowcharts we use in this book consists of a few symbols, namely:

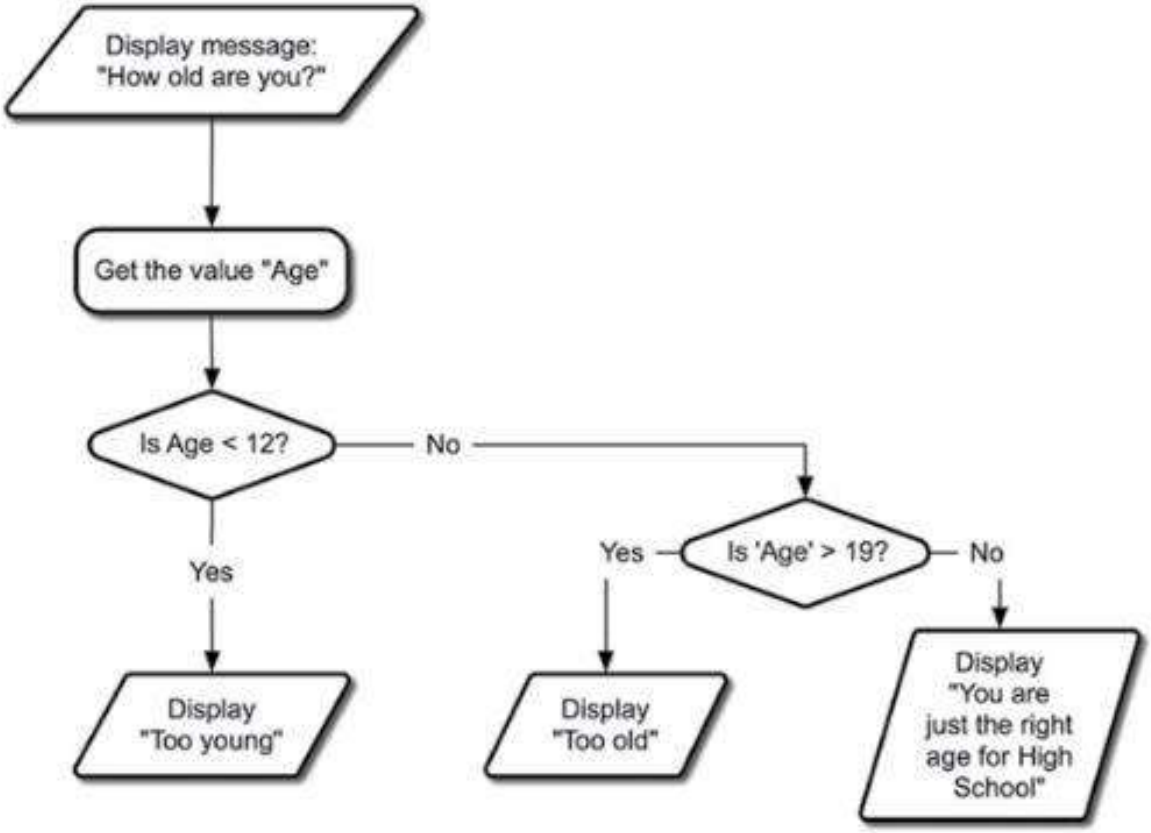
- **Input** and **Output**
- Processing
  - An **operation** which must take place or
  - A **decision**
- A **connector**

Consider the following example:



This flowchart shows a typical process of how a computer gets input (first telling the user what it wants) and then generates a display based on some decision making.

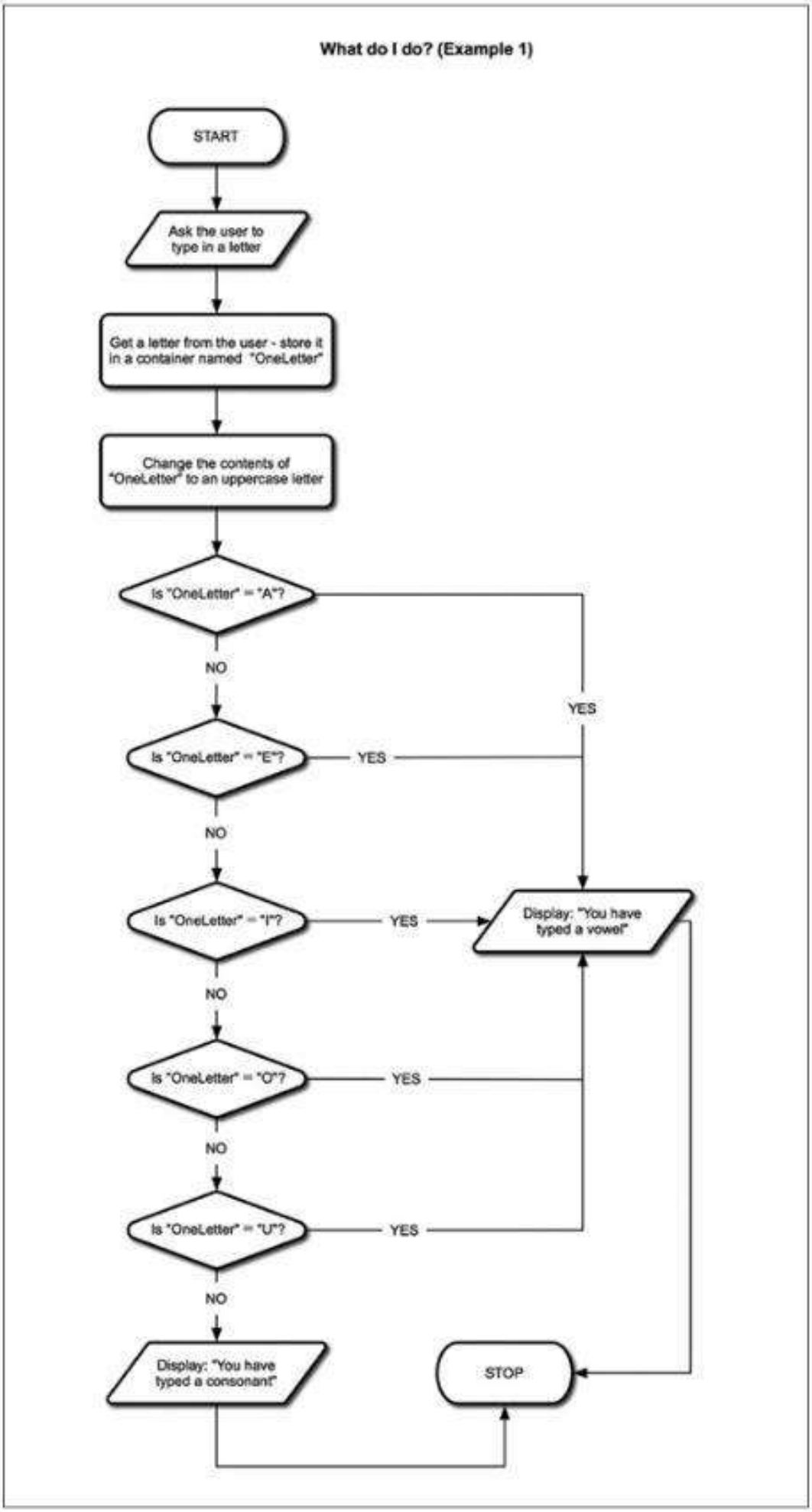
Example of a simple flowchart



We will use the flowchart tool throughout the course so that you can get used to the way that computers 'think' and do things. If you want to learn to program (and be good at it) then you have to learn to 'think' like a computer does!

1 Activity • Start to 'think' like a computer

1. Briefly describe the purpose of the process described by the 'Example of a simple flowchart' which was displayed previously.
2. What are the three main components of a flowchart?
3. What type of question must be asked to be able to make decisions like a computer does?
4. Examine the flowchart 'What do I do? (Example 1)' and explain what it does.
5. Examine the flowchart 'What do I do? (Example 2)' and explain what it does.


What do I do? (Example 1)
START
Ask the user to type in a letter
Get a letter from the user - store it in a container named "OneLetter"
Change the contents of "OneLetter" to an uppercase letter
Is "OneLetter" = "A"?
NO
YES
Is "OneLetter" = "E"?
YES
NO
Is "OneLetter" = "I"?
YES
Display: "You have typed a vowel"
NO
Is "OneLetter" = "O"?
YES
NO
Is "OneLetter" = "U"?
YES
NO
Display: "You have typed a consonant"
STOP

What do I do? (Example 2)

ROTFL Cat says:

- All decisions you make when trying to 'think' like a computer have to be made with questions that have only a 'Yes' or a 'No' as possible answers.
- Not all flowcharts have to include decisions.

Once you have studied the complete Module 1.1 (Thinking for a computer), you will start to learn a programming language called Scratch*. It can be downloaded for free from the Internet. You will write a number of programs using Scratch where you will have to 'think' like a computer, and first design your solutions using flowcharts, before the fun can begin.

We hope that you will enjoy the subject, learn a lot and have lots of fun!

The Study Opportunities Team

* Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu

Section 1

Algorithm design and programming

Modules

Module 1.1 Thinking for computers

Module 1.2 Introduction to graphical programming

Module 1.3 Input, processing, output

Module 1.4 Decision making

Module 1.5 Interaction between sprites

Module 1.6 Lists

Module 1.7 Prevent and find errors

Module 1.8 Projects

MODULE 1.1

Thinking for Computers

In the introduction you learnt a little about using a tool called a flowchart. It enables you to represent the steps involved in solving a problem in a visual way which is easy to read and understand. You also learnt that solutions to problems could involve making decisions, based on questions that only have 'Yes' or 'No' as an answer.

It is important to read the introduction of the book before starting this module.

In this module we are going to teach you how to approach a problem and divide it into tasks, so that eventually you will be able to use a computer as a tool to perform the tasks and solve the problem.

Computers and 'thinking'

Before we even start learning about problem solving there is one very important fact you have to know about computers:

Computers cannot think! Computers can only follow instructions.

That's a bit of a shocker! If computers can't think, then how can they beat us at games? Super computers (really large computers made up of many processors all working together) have beaten the best humanity has to offer in complex thinking games like chess *(Deep Blue* did that) and even at trivia games like Jeopardy *(Watson* did that). These are computer games that surely need the ability to 'think'?

The secret lies in the **sets of instructions** that the computer receives and executes. These sets of instructions are called **computer programs** and they are written by people.

It is therefore due to the creativity and amazing thinking skills of the human mind that instructions for these games can be compiled for computers to execute – creating the illusion that computers can 'think'.

Computers are fast, can work with large amounts of data and don't get bored or tired or need to sleep, eat or use the bathroom. They can perform millions of calculations in a fraction of a second. These characteristics of computers further strengthen the illusion that computers can 'think'.

Before we go on looking at how to compile sets of instructions called computer programs, it is worthwhile to explore the concept of computers and their ability to 'think'.

1 Activity • Explore Computers and their ability to 'think'

You need Internet access for this activity. The purpose is to make you think - and possibly even inspire you to follow a career in working to make 'thinking machines' a reality.

The activity can be done in groups in the format of a competition or a quiz.

Read the questions carefully and look for key words to search on.

1. Complete the statement: The term for machine thinking or intelligence is ...

2. What is the name of the test for computer intelligence?

3. Briefly explain what a computer has to be able to do to pass the test (mentioned in question 2) for computer intelligence.

4. Lotfi Zadeh made a contribution to teaching machines to make decisions the way humans do. What is that contribution called?

5. In his famous 'I, Robot' series of novels, Science Fiction author Isaac Asimov created the Three Laws of Robotics that must govern the behaviour of every robot. What are these laws?

6. What natural human fears are these laws (mentioned in question 5) designed to prevent?

7. Ray Kurzweil writes about an event called 'The Singularity'. Compile a brief summary explaining this concept.

8. Compile a summary / definition of the concept 'Neural Networks'.

9. What is a neuron and where is it found?

10. What kind of programming allows a computer to develop its own 'best solution' and is based on working like evolution, but speeded up to computer speeds?
'_____ is a way of solving problems by mimicking the same processes mother nature uses. They use the same combination of selection, recombination and mutation to evolve a solution to a problem'.

Create a solution for a problem

The process of learning how to instruct a computer to solve a problem, starts with first compiling the solution yourself, and then formulating it in a step-by-step way. This step-by-step solution is referred to as an algorithm. In other words, an algorithm is an instruction manual or a recipe explaining how to perform a task.

An **algorithm** is a sequence of steps that can be followed in order to solve a specific problem.

Algorithms in everyday life

We are executing algorithms almost every second of the day – when we answer the phone, when we eat, when we make coffee or tea, tie shoelaces, etc. Most of these everyday activities follow a sequence of steps that you learnt and that you now apply daily without even thinking about it. Some tasks, such as how to bake bread, how to install a specific software program or how to fix a printer can be more complex than others. However, a well-defined set of instructions can make these complex tasks easy to do!

2 Activity • Exploring characteristics of algorithms

In this activity you are going to explore the characteristics of a well-defined set of instructions (a good algorithm):

Group activity: Compile a simple set of instructions that can be used to explain to someone how to get into a word processor on a computer (or any other specific small task that everybody can relate to). Write the steps down in the order they should be executed.

Each group will read their solution (or report in another way). The class has to be extremely critical and identify the best solution. While you are busy evaluating the solutions, write the different reasons why some solutions were not accepted, on the board. Use these reasons to compile a list of characteristics of a good algorithm.

Some characteristics of a good algorithm are:

- One activity per step.
- Clear instructions (no ambiguity).
- The algorithm should have a clear beginning and end.
- Completeness – all the steps should be formulated as part of the solution.
- The steps should be in the correct order.
- All the steps should be related to the task to be completed – no unnecessary steps or information.
- The algorithm should be efficient. Make use of decisions and repetition to make the algorithm as short as possible.

You will learn more about decision making and repetition later in this module.

Presenting algorithms

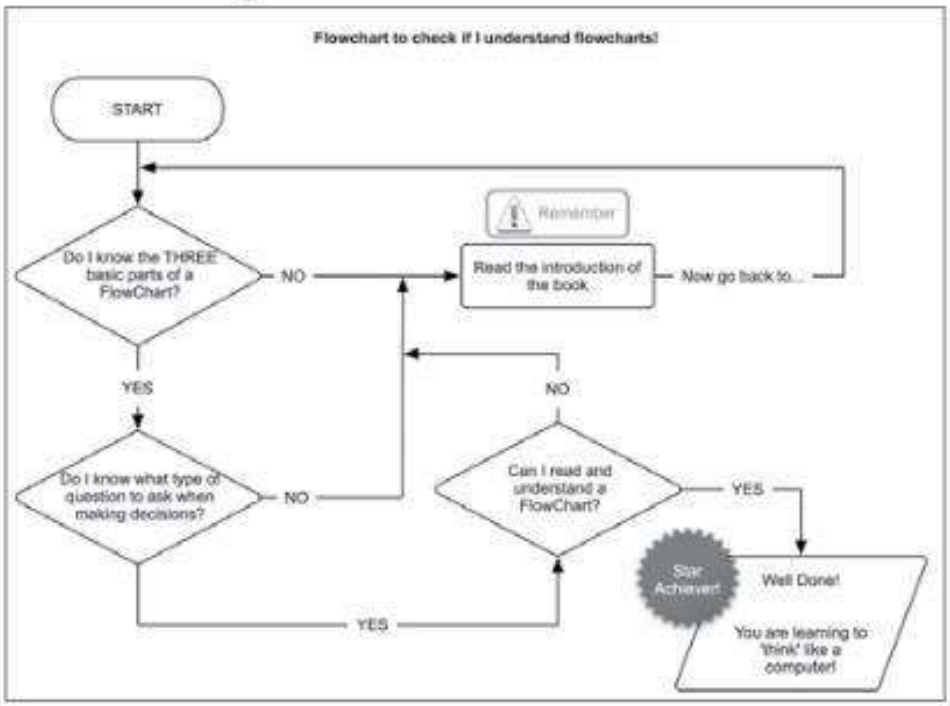
There are different ways to represent an algorithm. We have already introduced you to flowcharts in the introduction of the book. Other popular ways of representing algorithms are block diagrams and pseudo code. In this textbook we will use flowcharts and pseudo code to present algorithms.

Flowcharts

Let us revise your knowledge of flowcharts.

3 Activity • Revision – The basic concepts of flowcharts

Consider the following flowchart:

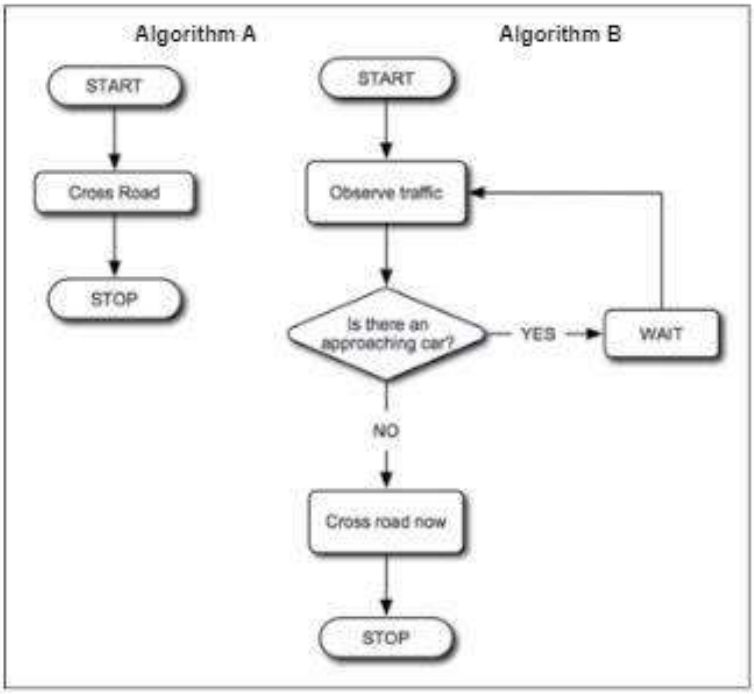


1. Identify the symbol indicating the flow of events in a flowchart.
2. Which shape is used to indicate decisions in a flow chart?
3. Identify the shape that contains an instruction that should be executed (not a decision).
4. Which part (steps) of the flowchart will be repeated when the answer to any one of the questions in the decision blocks is 'No'?
5. Describe in words all the conditions that have to be 'true' to reach the final step of this flowchart

4 Activity • Explore the use of flowcharts

Algorithm: Cross the road

Explain why Algorithm B is a better solution than Algorithm A. Make use of the list of criteria you compiled in Activity 2.



Notes on Activity 4:

- Algorithm A is unsuccessful because you cannot simply cross the road! We can argue that the step 'Cross Road' is not well defined.
- Algorithm B is more successful because you have to make sure that it is safe to cross the road first.
- A decision block has been added to test a specific condition.
- The content of a decision block is always a question with either a 'Yes' or a 'No' as an answer.
- Repetition takes place when the answer to the question in this particular decision block is 'Yes'.
- When the answer is 'No', the condition has been met, and the person can cross the road.
- The flowchart stops once the person has crossed the road.

Pseudo code

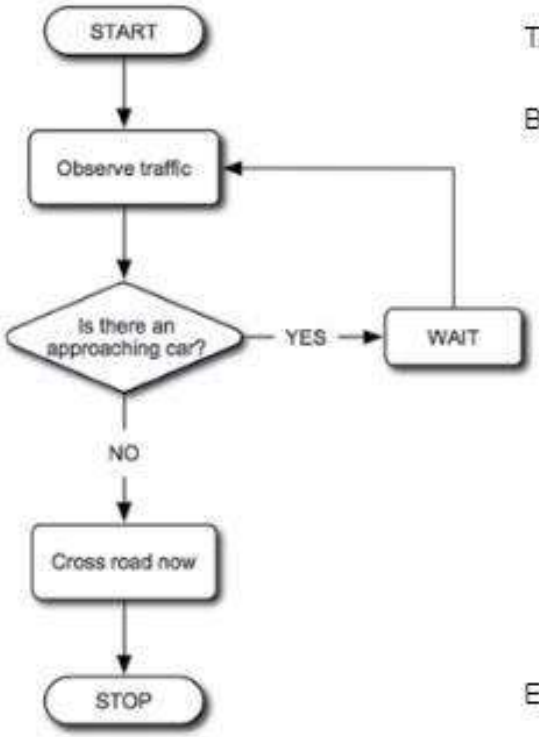
Pseudo code is a structured language-based way of presenting an algorithm, similar to the methods of recipes in a cookbook – short and clear step-by-step instructions.

Some basic guidelines to compile pseudo code are the following:

- Use short, clear instructions.
- Clearly indicate the start and the end of the task to be executed.
- Use variable names to indicate input, processing and output.
- Use keywords to indicate decision making and repetition, for example:

  While . . . Do . . . EndDo
  If . . . EndIf
  Repeat . . . Until
- When the order in which instructions will be executed 'branch out', for instance, when a decision or repetition is made, all the instructions in the 'branch' must be indented.

Let's see how Algorithm B from Activity 4 will be represented in pseudo code.




```
TASK: Cross the road

BEGIN

    Observe traffic

    While traffic approaching Do

        Wait

        Observe traffic

    EndWhile

    Cross the road

END
```

ROTFL Cat says:

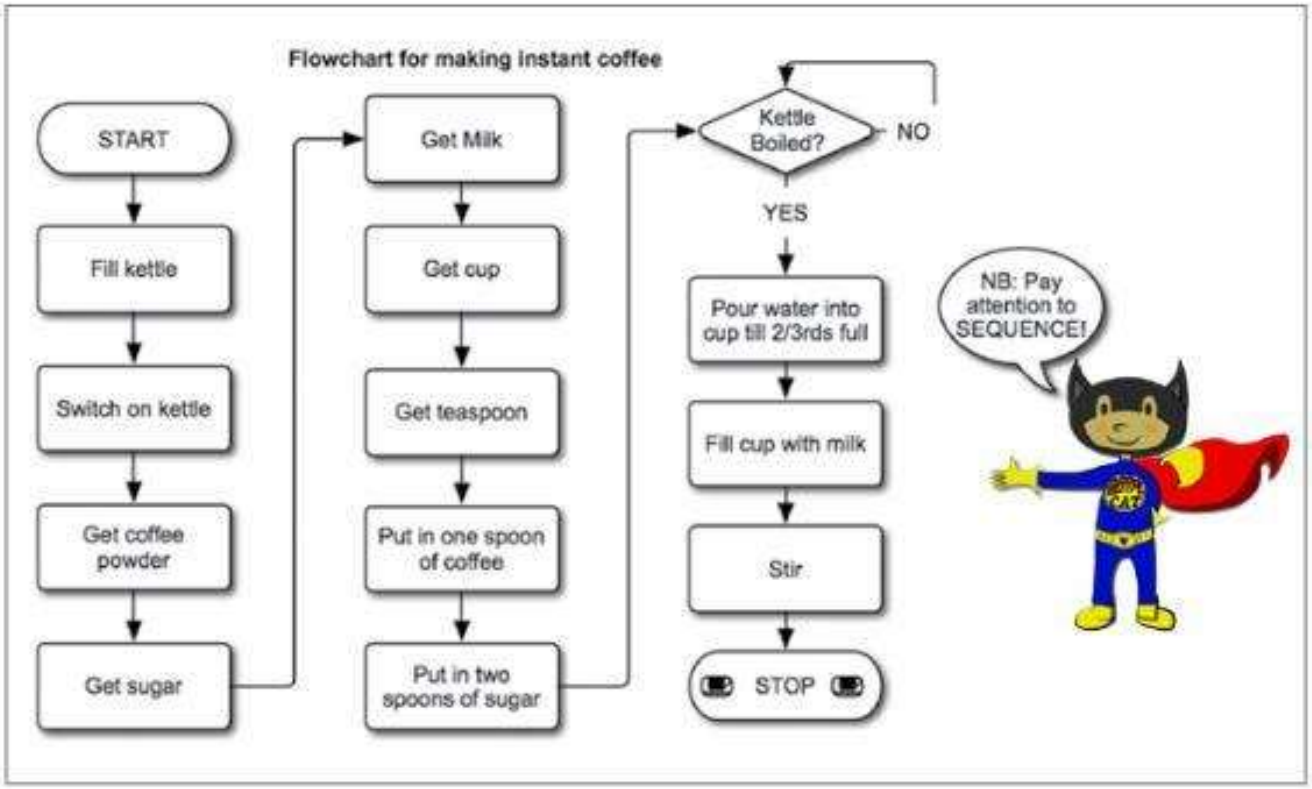
There are no fixed rules for the way to write pseudo code. In the end a programmer must be able to convert the pseudo code to code for a specific programming language. Usually a team of programmers agree on their own rules for pseudo code based on the basic guidelines given above.

1. Take a look at the map in the

Concrete and abstract algorithms

How to make coffee: concrete and abstract algorithms

Consider the following algorithm on how to make coffee:



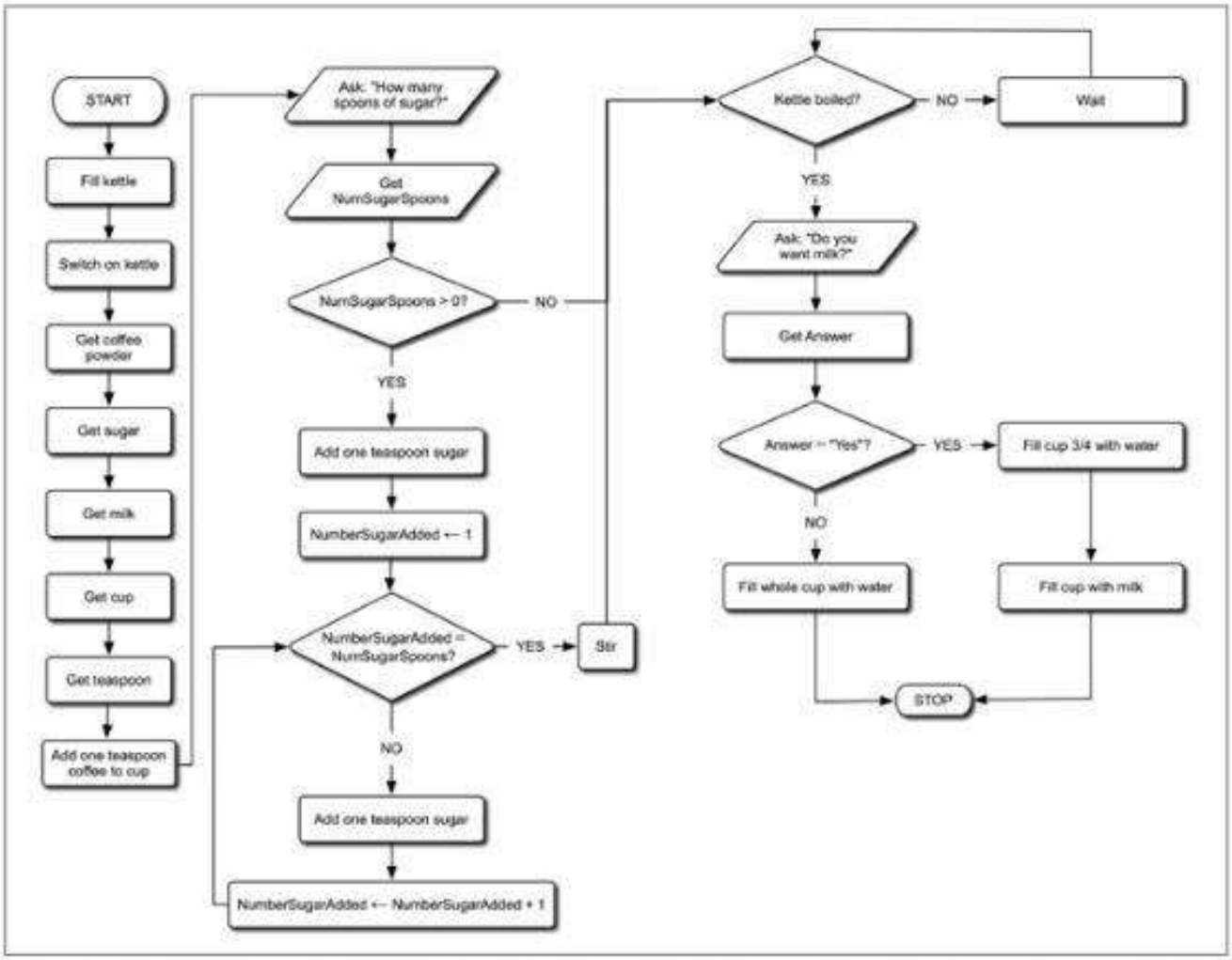
The instructions contain specific information regarding milk and sugar, and can only be used to make coffee with milk and two teaspoons of sugar. If a person wants coffee with one teaspoon of sugar, and no milk, you will have to write another algorithm using the new requirements. We refer to an algorithm written for a specific situation with specific requirements as a **concrete algorithm.**

However, when you write a few concrete algorithms performing the same task (e.g. making coffee) but using different data (measurements) each time, you will discover that you have to write the same basics steps for all the algorithms. It would be more effective to develop a 'general' type of algorithm containing the basic steps to solve a problem, but with no concrete user-specific information. In this case, the same algorithm can be used to perform the task many times – each time with different information according to the requirements of the user. Look at the flowchart on the following page.

This flowchart provides for interaction with the user by asking questions (requesting input) of which the answers can differ from user to user. The values which are user-specific are stored in variables (in the same way that variables are used in mathematical equations). The content of a variable is unknown while the algorithm is developed. Only when the algorithm is executed, actual values are assigned to the variables – until then, variables are used to 'keep space' for the actual values.

In this flowchart, one of the variable values is the number of teaspoons of sugar. We used the name NumSugarSpoons as the name of the variable.

This solution is more **abstract** and can be used to make a good cup of coffee for most people since the amount of milk and sugar were not specified beforehand as with concrete algorithms.



How to determine if a number is an even number: concrete and abstract algorithms

Another example of the difference between a concrete and an abstract algorithm is shown in the following table:

Concrete	Abstract
How do we determine whether 6 is an even number?	**How do we determine whether any number is an even number?**
We ask: Is 6 / 2 = 3 with no remainder? If the answer is 'Yes', 6 is an even number	We ask: Will the calculation Number / 2 = a result with no remainder? If the answer is 'Yes', Number is an even number.
The solution can only test the number 6.	The solution can test any number. A 'place holder' is used instead of a specific number. We refer to the 'place holder' as a variable.
Use: Every time we want to test another number we have to design a new algorithm, specific to the new value. This is not very efficient.	Use: We do not have to design a new algorithm every time we want to test another number. We can use the same algorithm. Abstract solutions are therefore more efficient, because they can be used for any value instead of for only one value..

ROTFL Cat says:

The key difference between concrete (limited) algorithms and abstract (general) algorithms is the use of variables.

In Maths we usually use variables such as x and y, but when writing algorithms (and later computer programs), we use descriptive variable names which give an indication of the value it will store.

Four steps to follow when working out an algorithm

Step 1: Understand the problem

- Formulate the problem as clearly as you can (it helps to state it in your own words). What is the purpose of the solution? You have to be sure of what you want to achieve.
- Determine what output is required.

Step 2: Analyse the problem

- Investigate important concepts and requirements. This involves asking a lot of 'What?' questions.
- Determine what data and information is given.
- Work out how you would solve the problem yourself, without using the computer. This involves asking a lot of 'How?' and 'When?' questions.
- Determine what data (possible input) is required.

Step 3: Create a solution

- List the steps to solve the problem in the correct sequential order.
- If possible, break down the steps so each one signifies only one task.
- Ensure the sequence of steps is correct.

Step 4: Test the solution

- Test your solution by using concrete data.

Algorithms for simple mathematical problems

An algorithm to calculate the area of a square

Let us see how we can teach the computer to calculate the area of a square by

- setting up an algorithm (applying the four steps) and
- representing it using pseudo code.

Step 1: Understand the problem	
Questions	**Answers**
What problem needs to be solved? (Describe the problem in your own words.)	Work out the area of a square.
What output is required?	The area of a square.
Step 2: Analyse the problem	
Questions	**Answers**
What is a square?	A shape with sides of the same length.
How do I calculate the area of a square? (What is the formula to calculate the area of a square?) What data is given?	Area = Length of the sides x Length of the sides No concrete data
How would I solve this problem without the computer? (It helps to use concrete values to understand the formula better.)	Get the length of the side of a square (input) Use the formula to calculate the area Area = Length x Length = 5 x 5 = 25
What data will I need as input?	A number that indicates the length of the sides of the square.

Step 3: Create a solution	
Get the input (Length of the side of the square) Calculate the Area (Area = Length x Length) Display the calculated Area	
Step 4: Test the solution	
Test data: Input: Length of 6 Test data: Input: Length of 7 Test data: Input: Length of 55	Expected output: Area is 36 Expected output: Area is 49 Expected output: Area is 3025

Once you understand the problem yourself and know the steps to the solution, you have to work out how to present the steps to the solution (the algorithm).

Pseudo code

TASK: Calculate the area of a square.
BEGIN
    Ask 'What is the length of the side of the square?'
    Get Length
    Area = Length x Length
    Display 'Area is:', Area
END

An algorithm to provide a quote

Problem statement: A person lays floor tiles at a tariff of R30.45 per m². Calculate the total cost to tile a room (a rectangle). The person wants a deposit of 10% of the cost prior to commencing, and the balance on completion of the work. Calculate and display the final amount, the deposit, as well as the balance that has to be paid on completion of the work.

Step 1: Understand the problem	
Questions	**Answers**
What is the problem that needs to be solved? (Describe the problem in your own words.) What output is required?	Calculate the total cost to tile a room (a rectangle). The final cost, the deposit and the balance.
Step 2: Analyse the problem	
Questions	**Answers**
How do I calculate the area to be tiled?	Area of a rectangle = Length x Width
What data is given?	Tariff of R 30.45 per m² Deposit is 10% of cost
How would I solve this problem without the computer? Use concrete values to test formulae and calculations.	Get the length and width of the room. Use the formula to calculate the area Area = Length x Width = 5 x 3 = 15 Cost = Area x Tariff = 15 x 30.45 = 456.75 Deposit = 10/100 x Cost = 45.68 Balance = Cost – Deposit = 456.75 – 45.68 = 411.07
What data will I need as input?	The length of the room The width of the room

Step 3: Create a solution
Get the length and width of the room (input) Calculate the area (Area = Length x Width) Calculate the Cost (Cost = Area x Tariff) Calculate the Deposit (Deposit = 10/100 x Cost) Calculate the Balance (Cost – Deposit) Display Cost, Deposit, Balance (output)
Step 4:Test the solution
Test data: Input 3 (Length) and 2 (Width) Output: Cost: R 182.70 Deposit: R 18.27 Balance: R 164.43

Pseudo code

```
TASK: Calculate costs to tile a rectangular room
BEGIN
    Ask 'Length of the room?'
    Get Length
    Ask 'Width of the room?'
    Get Width
    Area = Length x Width
    Cost = Area x 30.45
    Deposit = Cost x 0.1
    Balance = Cost - Deposit
    Display 'Cost is: R', Cost
    Display 'Deposit is: R', Deposit
    Display 'Balance is: R', Balance
END
```

Activity • Create algorithms for simple calculations

1. Create a paper-based algorithm file
   - Get or create a file and label it **My Algorithms.**
   - Either photocopy, print out or re-draw the algorithms we have looked at so far.
   - Make sure that each algorithm has a suitable heading to describe what it does.
   - File the algorithms alphabetically.

   As you advance through the year you will add printouts of algorithms to this file. Eventually you will have a collection of algorithms that you can refer to and apply to problems that you need to write programs for.

   You are welcome to create your own electronic copies as well, but having printouts can actually help you to study.

2. Do the following for each of the given algorithms:

(i) List the variables used in each algorithm.

(ii) Indicate the value for each variable when testing the algorithm using the test data provided.

(iii) Give the algorithm a suitable heading based on its task.

(iv) Give a suitable message to be displayed with the output.

a) TASK:

```
BEGIN
    Ask 'What is your ID number?'
    Get IDNumber
    Ask 'What is the current year? (YYYY)'
    Get CurrentYear
    FirstTwo = First two digits of IDNumber
    Num1 = 1900 + FirstTwo
    Num2 = CurrentYear – Num1
    Display Num2
END
```

Test data	
IDnumber	CurrentYear
6902080005084	2012
9023120004087	2014

b) TASK:

```
BEGIN
    Ask  'What is the area of the room?'
    Get AreaOfRoom
    TheNumber = INTEGEROF(AreaOfRoom/4) + 1
    Display TheNumber
END
```

Test data
AreaOfRoom
25
40
23

c) TASK:

```
BEGIN
    Ask 'How many sweets do you have?'
    Get NumberOfSweets
    Ask 'How many children are there?'
    Get NumberChildren
    Number1 = INTEGEROF (NumberOfSweets / NumberChildren)
    Number2 = NumberOfSweets – (NumberChildren x Number1)
    Display Number1
    Display Number2
END
```

Test data	
NumberOfSweets	NumberChildren
20	6
30	5

3. Group activity: Create algorithms to do simple mathematical calculations

- Split the class into groups of 4.
- Each group will be given a problem from the list below to create an algorithm.
- Each group must present and explain their algorithm (flowchart or pseudo code) to the class.
- Discuss the suggested algorithms with one another and add copies of all algorithms to your algorithms file.

Choose any of the following problems

a) Work out the percentage for a test mark. The maximum mark as well as the mark achieved are available as input.

b) Calculate the amount each person must pay as a part of a group bill tip at a restaurant. Tips are usually around 10% of the bill, but the algorithm must provide for a value, indicating what percentage of the bill the group wants to give as a tip, to be entered. The bill amount and the number of people in the group must also be entered as input.

c) Work out the fuel consumption of a car in litres per 100 km if the amount of fuel used and the distance travelled are required as input.

d) Athletes measure the time it takes to complete a marathon in minutes. Calculate the number of hours and minutes run by the athlete. (For example: 134 minutes is 2 hours and 14 minutes.)

e) Determine the amount of money in Pounds that a person can buy. You are given the exchange rate, as well as the amount, in Rand, that he or she wants to spend.

f) Determine the amount of money in Rand that a person can buy. You are given the exchange rate, as well as the amount, in Pounds, that he or she wants to spend.

g) One carton of tiles can cover an area of 2.4 m2. Calculate the number of cartons necessary to cover a room with a certain surface area. Only full cartons can be bought. If you need 4.2 cartons of tiles, you would then have to buy 5 cartons.

h) You want to add a trimming around the edge of a round (circle-shaped) carpet. The price of the trimming is R 15.00 per meter. You also want to know what area of floor the carpet will cover (without the trimming). Display how much trimming has to be bought, the total cost of the trimming and the area of the floor that the carpet will cover (without the trimming).

Hint:

Calculate the circumference of the carpet to determine the length of the trimming
Formulae: The circumference of a circle: $2\pi r$

The area of a circle: $\pi r2$

π is the symbol for 'Pi, and Pi has the value 3.14159265

r refers to the radius of the circle.

Algorithms with decisions

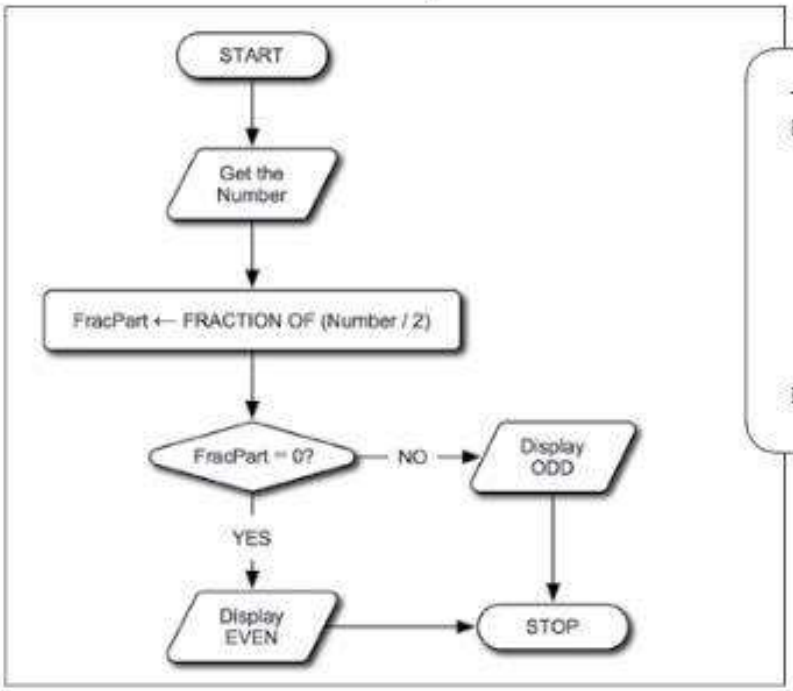
Sometimes the algorithm that you design for solving a problem will involve using more complex mathematical calculations. Now it becomes really interesting, because you have to use your basic knowledge of mathematics to solve a problem, and then compile the algorithm for the solution!

Problem statement: Design an algorithm to determine whether a number is even or odd.

Step 1: Understand the problem	
Question	**Answer**
What is the problem that must be solved? (Describe the problem in your own words.) What output is required?	Determine whether a number is even or odd. Display the word 'Even' if an even number was entered, or the word 'Odd' if an odd number was entered.
Step 2: Analyse the problem	
Question	**Answer**
What are the characteristics of an even number?	A whole number which, when divided by 2, has a fraction part of 0.
What is known or given?	No additional information is given.
How would I solve this problem without the computer?	Divide the number by 2. If the answer contains a fraction part equal to 0, the number is an even number.
Use concrete values to test the proposed solution.	10 / 2 = 5.0 (the fraction part is 0 so it is an even number) 11/2 = 5.5 (the fraction part is ≠ 0, so it is an odd number)
What data (input) will I need to solve this problem?	Any whole number.
Step 3: Create a solution	
1. Get the number. 2. Determine the fraction-part of the answer when the number is divided by 2. 3. If the fraction part ≠ 0 display 'ODD' 4. If the fraction part = 0 display 'EVEN'	
Step 4: Test the solution	

Test data: 5 Input: 5 Processing: 5 / 2 = 2.5 The fraction part is 0.5 The fraction part is ≠ 0 Display 'ODD'	Test Data: 6 Input: 6 Processing: 6 / 2 = 3.0 The fraction part is 0.0 The fraction part is = 0 Display 'EVEN

Let's see how the solution can be represented:




```
TASK: DetermineEven
BEGIN
    Ask 'What is the number?'
    Obtain Number
    FracPart = FRACTION OF(Number/2)
    If FracPart = 0
        Display 'EVEN'
    Else
        Display 'ODD'
    EndIf
END
```

ROTFL Cat says:

ROTFL Cat offers tips on working with flowcharts:

- Always use straight lines joined at right angles for your lines.
- The lines indicate the flow of events. Always use an arrow as part of a line to indicate what should happen next.
- Input and output are represented by parallelograms.
- Input, output and process shapes have one incoming line and one outgoing line.
- Decision making shapes have one incoming line and two outgoing lines.

7 Activity • Algorithms with calculations and decision making

Create neatly designed flowcharts to solve the following problems. Also write pseudo code. Place copies of the algorithms in your algorithms file.

1. Determine whether a number is a factor of another number.
2. Calculate the percentage a learner gets for a test if the learner attained a certain mark and the total marks for the test are 60. Determine whether the learner has a distinction (≥ 80). Display the mark, percentage and suitable message if the learner has a distinction.
3. A manager gets paid double the waiter tariff per hour. Enter the number of hours worked, the worker tariff per hour and the code indicating the type of worker (W = Waiter, M = Manager). If the code is M (Manager) the tariff paid per hour must be doubled. Calculate and display the payment.

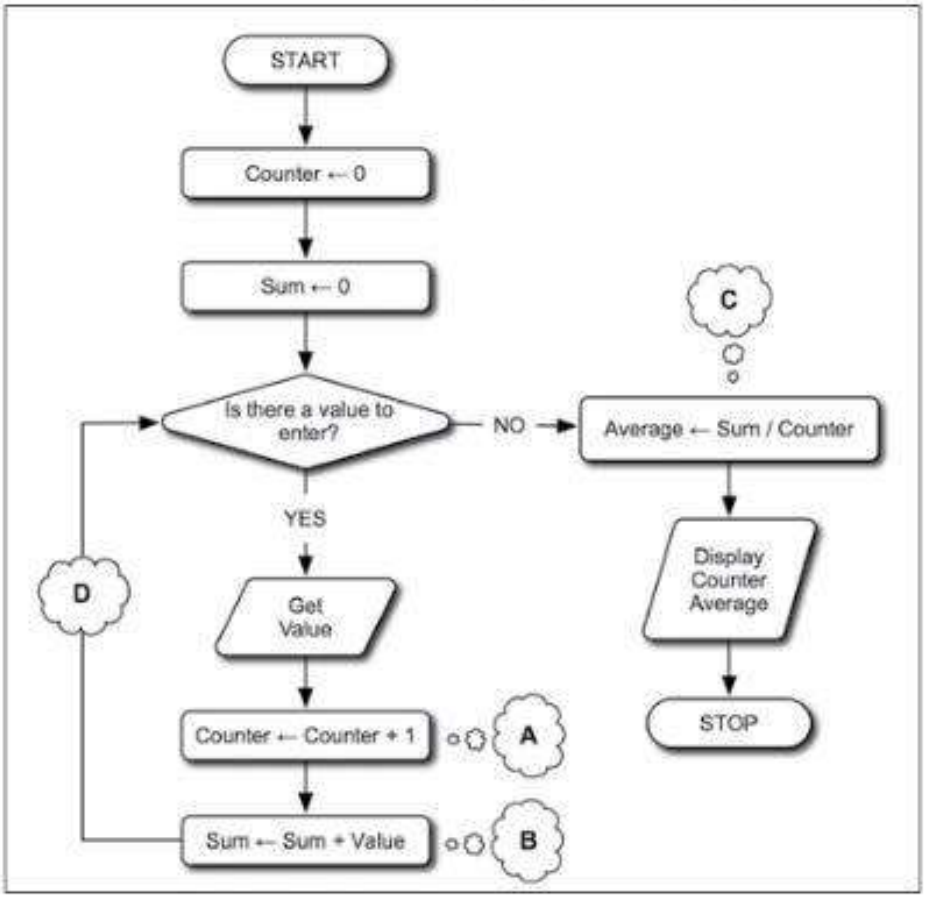
Algorithms with repetition

One of the most useful skills you can develop is to learn to recognise repetition in the steps to solve a problem. Repetition occurs when the same step(s) is executed more than once. For example, calculate the average mark of each the learner in the class. If there are 40 learners in the class, the same calculation has to be done 40 times!

We will now take a look at two algorithms that need repetition in order to solve the problem.

Summarising data from a series of numbers

Programmers often have to develop programs where the average of a number of values needs to be calculated – for example, programs which generate report cards for learners.



The following flowchart displays an algorithm to

- read a number of values
- count the number of values entered
- calculate the average of all the numbers
- display how many values have been read
- display what the average of all the values is.

8 Activity • Analysing a given flowchart

Analyse the flowchart in the previous section and answer the following questions:

1. What is the input to this problem?
2. What is the output?
3. Will it change the result of the algorithm if the instructions in the blocks marked A and B are swapped? (First calculate the sum, then increase the counter).
4. Will it be effective to move the instruction in the block marked C to position D?
5. If the user of this flowchart (and eventually the user of the computer program), indicates that there is no value to enter the very first time the decision block 'Is there a value to enter?' is encountered, an error will occur. The value of Sum will still be zero, so in block C a division by zero will be done, which is illegal. Adjust the flowchart so this problem will not be encountered.

6. The following pseudo code represents the algorithm to calculate and display the average of a few numbers (see the previous flowchart). The error described in question 5 will also occur when this pseudo code is executed. Indicate which instruction(s) in the pseudo code should be adjusted to ensure that the error will not occur. Rewrite the instruction(s).

```
TASK: Calculate and display the average of a few numbers
BEGIN
    Counter = 0
    Sum = 0
    While another number DO
        Ask 'What is the number?'
        Get Number
        Counter = Counter + 1
        Sum = Sum + Number
    EndWhile
    Average = Sum / Counter
    Display Counter
    Display Average
END
```

Note how the code in the repetition is indented.

Finding the smallest value in a series of numbers

The key word here is **series**. That tells us that we have a whole lot of numbers to work with. We need to work out which number in the series is the smallest.

Step 1: Understand the problem	
Questions	**Answers**
What is the problem to be solved? What output is required?	Find the smallest number from a series of inputs. The smallest number of all those numbers entered.
Step 2: Analyse the problem	
Questions	**Answers**
What is known or given?	No data is given other than the fact that there will be a series of numbers entered.
How will we keep track of the smallest value?	Use a temporary value 'Smallest' to keep the smallest number.
How would I solve this problem without the computer?	Get a number. If it is the first time you have received a number, write down this number – this is the smallest number you have so far. If it is not the first time you have received a number, compare the number to the smallest number. If the number is smaller than the smallest number, erase the smallest number and write down the number as the smallest. Repeat all the steps until there are no more new numbers to enter. Display the smallest number.
What data (input) will I need to solve this problem?	A series of numbers.
Step 3: Create a solution	
1. Counter = 0 (used only to determine when the first number has been entered) 2. Get the number 3. Increase Counter by 1 4. If Counter = 1, Smallest = number 5. If Counter <> 1 6. If number < Smallest, Smallest = number 7. Repeat steps 2 to 6 until there are no more numbers 8. Display Smallest	

Step 4: Test the solution

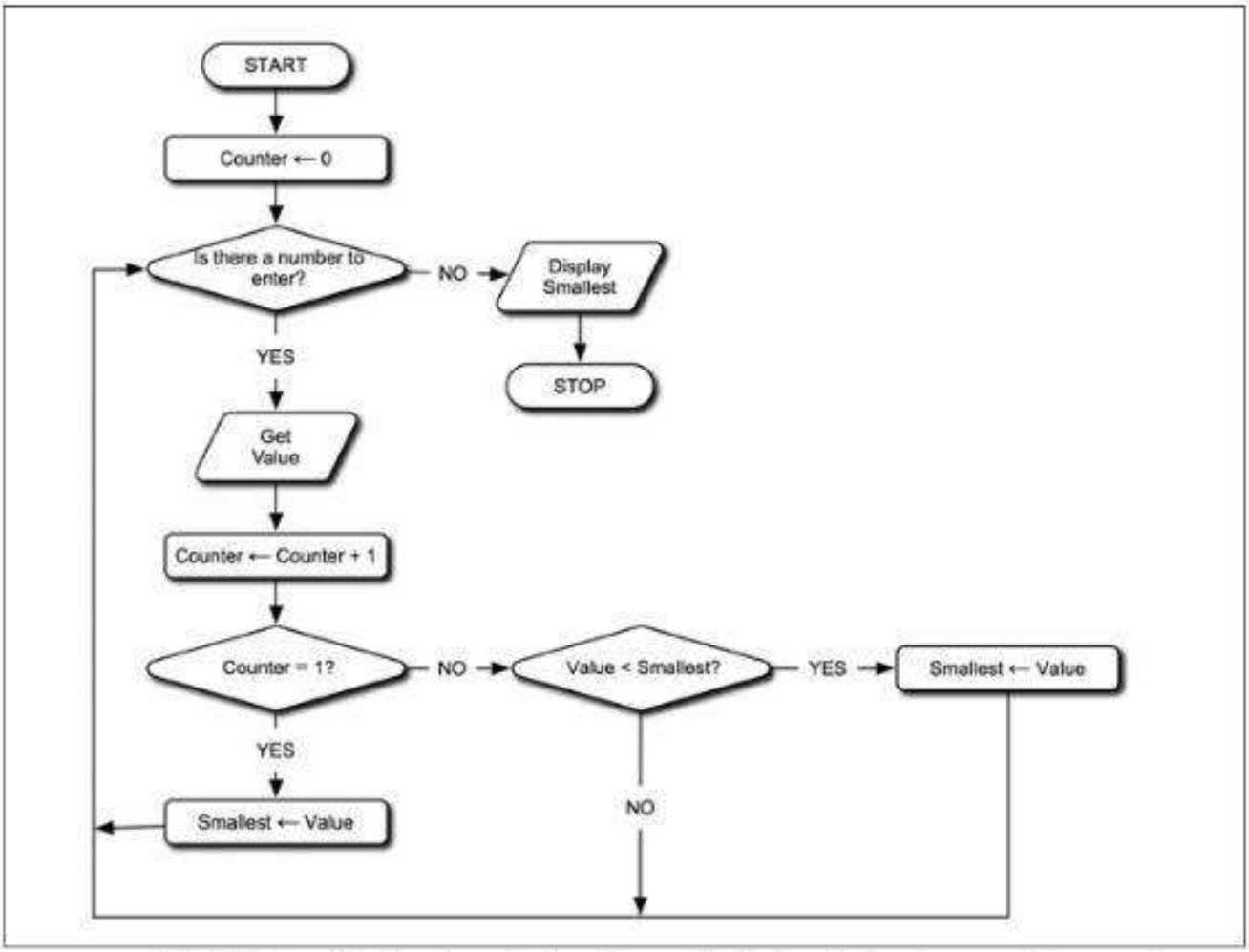
Use the series of numbers: 3; -6; 8
Counter ← 0
Get 3 as the first number,
Counter ← 1
Counter = 1, so Smallest ← 3
Is there another number? Yes

Get -6
Counter ← 2
Counter <> 1
-6 < 3, so Smallest ← -6
Is there another number? Yes

Get 8
Counter ← 3
Counter <> 1
8 > -6, so Smallest will not change
Is there another number? No
User indicates there are no more numbers

Display the value of Smallest: -6

Again – this kind of thinking and planning makes it quick and easy to create a flowchart to represent the solution.



Note: This algorithm has been designed to work for the input of positive as well as negative values.

Challenge: Re-write the algorithm to find the smallest value if all values will be greater of equal to zero.

9 Activity • Repetition in an algorithm

Create an algorithm for each of the following scenarios. Remember to use repetition in your algorithm.

1. Let the user guess a secret number. Count the number of guesses it took to guess the number correctly.
2. Display all the factors of a number.
3. Display the 6 x table.
4. Calculate the power of a number, e.g. 34, using a loop (you will need the base and the exponent).

ROTFL Cat says:

These algorithms will be tools that you use when you begin to write your own programs and you will need them in future IT studies as well.

- Pay attention to them.
- Know them well!
- Keep your file updated.

Glossary

Abstract solution	A solution which makes use of general concepts and variables in solving a problem.
Algorithm	A sequence of steps that can be followed in order to solve a specific problem.
Computer program	A set of instructions written in a specific computer programming language which instructs the computer how to solve a problem.
Concrete solution	A solution which makes use of fixed, specific values and steps in solving a problem.
Flowchart	A diagrammatic visual representation of the sequence of steps required to solve a problem.
Problem solving	Working out the steps required to reach a desired state based on a given set of starting values.
Pseudo code	Pseudo code is a structured, language-based way of presenting an algorithm, similar to the methods of recipes in a cookbook – short and clear step-by-step instructions.

MODULE 1.2

Introduction to graphical programming

Scratch is a programming language for learning the basic concepts that you use when writing a computer program. It is very visual and it is easy to write programs which make use of colour, animation and sound.

ROTFL Cat says:

Don't just learn how to do things in Scratch – think about what you are doing and how you are doing it – these concepts stay the same in every programming language!

Learning about a computer program is little or no fun if you have to wade through pages of dreary instructions and screenshots in a textbook. We are sure you will learn to use the necessary skills much faster by seeing examples of programs (either on videos or hard copies) and then applying the principles when creating your own programs.

Gather your resources

There are a number of documents and files you need to access when you learn Scratch and work though the modules in this book. Here is a list of everything you need to collect before you continue working:

Source	Where to find it
ScratchReferenceGuide (a PDF file)	• Open the Scratch program. • Click on **Help**, **Help Page**, **Reference Guide**. You can save the pdf file on your computer's hard drive and review the electronic copy, or make a printout. 
Scratch's Help Screens	Open the Scratch program. Click on **Help**, **Help Page**, **Help Screens**. A web page will be displayed containing information on each program block.
Scratch is gr8! Help file Study Opportunities' help facility. (A file with links to videos and presentations which demonstrate and explain various concepts.)	It is available on the DVD which is supplied with the Teacher's Guide.
Sprites, costumes and **backgrounds**	You will be using this in your own programs. It is available on the DVD which is supplied with the Teacher's Guide.

1 Activity • Overview of Scratch and first program

Watch the following videos from the Scratch is gr8! Help file: Overview of Scratch; Working with sprites; The stage; Using program blocks.

Scratch is easy and fun to use because

- it presents you with immediate feedback in a graphical environment.
- the computer instructions looks like building blocks – so writing a program is like building a puzzle.
- it gives easy access to complex tasks such as graphics, animation and sound.

Get to know the Scratch environment

Scratch lets you do drag-and-drop programming. This means that you don't have to learn and type a whole lot of commands and remember what they do. Instead you are given commands that are shown to you visually in blocks in a way that helps you to understand what they mean and do. You simply drag and arrange these commands to create your program.

Briefly, the Scratch interface is divided into three areas, namely:

- On the left of the screen are the instructions (program blocks) you can use to write your program, all arranged in categories that describe what they are used for. This is called the **Blocks Palette**.
- In the middle of the screen is the area where you 'write' your program by dragging the program blocks from the **Blocks Palette** and making sure they connect to each other (like jigsaw puzzle pieces). This is called the **Scripts Area**.
- On the right is where you see a **stage** where all the action happens when you run your program. There is also an area where you can choose different sprites or actors to carry out your instructions (the **Sprite List**). A program can involve more than one sprite and each sprite can have its own set of instructions (or script).

Tips for saving your work

- We recommend that you save your programs often, and save each program with a name *before* you start creating the program. If you are creating a new program by changing an old one, you need to use the **File, Save As** option before making any changes.
- Create a folder for your Scratch programs.
- Decide on your own naming convention. For example you can use the name Mod1_1_Act2_Costumes to save your first Scratch program, Mod1_1_Act2_ProgramBlocks to save your second Scratch program and so on.
- Use descriptive names so that you can find a specific program easily through its file name.

2 Activity • Exploring the Scratch interface

For each of these exercises create a new Scratch project, and then follow the steps until your program matches the screenshot.

Remember to consult your resources if you are unsure of the way to complete a task.

1. Costumes & Sprites

   The purpose of this activity is for you to practice to change a sprite's costumes and add new sprites to the stage.

a) Delete Costume2.

b) Change Costume1 so the orange is red (the Fill tool works well for this).

c) Make a copy of Costume1 (it will automatically be called Costume2) and flip it around horizontally (you need to Edit it to do this).

d) Import a new costume - the cartoon car (Car1 in the Transportation folder).

e) Copy Costume2 four times and rotate each so it matches the directions of costumes 3 to 6 in the screenshot.

f) Edit Costume2 to change the mouth and eyes to look like the screenshot.

g) Use the **New Sprite** option, and add Sprite2 and Sprite3 as in the screenshot.

h) Position the sprites on the stage by dragging them around until they match the screenshot.

2. Adding and removing program blocks

The purpose of this activity is for you to drag program blocks onto the Scripts Area. You will also practice how to add and remove program blocks from an existing script.

a) Start a new project.

b) Import the pencil costume for Sprite1.

c) Delete the other costumes for Sprite1.

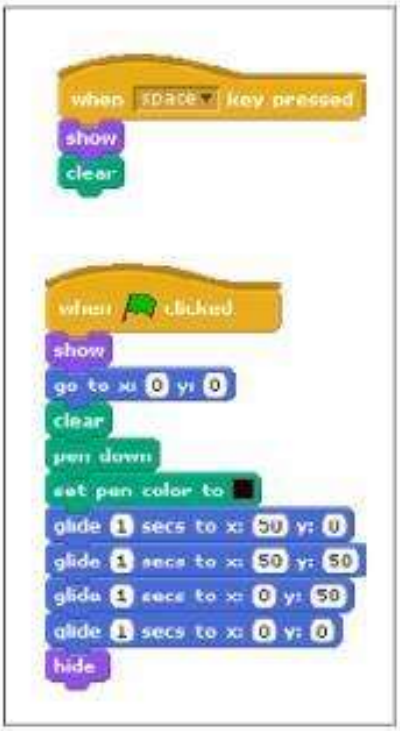
d) Create the scripts for Sprite1 by using the instructions from the scripts in the screenshot. Please note:

- It is important that you put the instructions in the same order.
- You will find the instructions you need in the **Control**, **Pen** and **Motion** instruction groups.
- Make sure that the instructions are joined (they behave like magnets and click into each other).

e) Run the program (click the green flag icon above the stage).

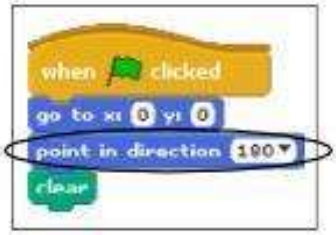
f) Describe to the person next to you what the program is doing.

g) Cut the program blocks **[show]** and **[hide]** (use the scissors on the toolbar). Run the program and explain the purpose of the two blocks you cut to the person next to you.



h) Add the program block as indicated in the screenshot.

i) Experiment with different values of the degrees the sprite should point in.



3. Changing the background of the stage

The purpose of this activity is for you to change the costumes of the stage and ensure you are able to find different backgrounds. To illustrate this we will build a simple photo album:

a) Start a new project.

b) Delete Sprite1 .

c) Click on the **Backgrounds** tab.

d) Import at least 5 different backgrounds.

e) Delete the original white background.

f) Create the following script:

When you run this application you will get a new background every time you click the green flag.

ROTFL Cat says:

The following terminology is important:

Program blocks (instructions / code) are used to write a **script** for a sprite.

The unit of blocks creating the script is called a **stack**.

Blocks rounded at the top are called **Hats**.

Blocks which can click into one another are called **Stack Blocks**.

A hat is always placed at the top of a stack.

A hat always waits for an **event** to take place.

An event can be one of many different actions:

- Click on the green flag.
- Push the space bar.
- Push a key on the keyboard.
- Click on a sprite.

The same sprite can react to more than one event.

We refer to a script or to the project (which can contain many sprites with many scripts) as a **program**.

We refer to the sprites as **actors**.

Teach a sprite to perform basic tasks

You are now going to create another brief program and we will discuss in detail what each instruction in the program does. You can run the program and review the explanations until you are sure you understand each instruction.

3 ACtivity • Create a more complicated program

a) Start a new project.

b) Load ROTFL Cat as a costume for Sprite1.

c) Delete the default costumes for Sprite1.

d) Load the pencil as Costume2 for Sprite1.

e) Move Sprite1 to the bottom left corner (see screenshot).

f) Click on the Stage icon in the Sprite List area of the screen.

g) Load (import) Chalkboard as the background for the stage.

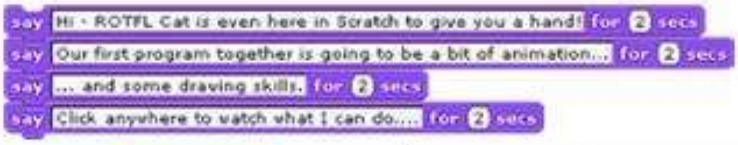
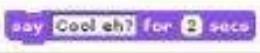
h) Click on Sprite1 again and go to the Sounds tab.

i) Load the sounds Pop, DripDrop and MotorCyclePassing from the default Scratch sound folders (look in all the folders till you find the sounds – this will give you a chance to experiment with all the sounds).

The environment for the program has now been created. Now you must write the program itself by inserting each program block in the correct position as indicated in the program which follows. Look through the instruction groups to find the instructions you need. (Remember that all the program blocks should fit into one another – there should be no spaces in between blocks. The blocks in this book only have spaces so you can see clearly which explanation is linked to which blocks.)

Pay attention to:

- The way you have to set the stage and actors (sprites) up (clear the screen, make the ROTFL Cat sprite large, pick up the pen, etc.) before you start the real program.
- The use of **[say]** and **[think]** instructions to communicate to the user what is happening.
- The use of sound which makes the program more entertaining.
- The importance of sequence. If anything is done out of order the program will not do what it should.

Scratch program	Explanation of the program
	Every program needs to start somewhere. This program starts when the green flag is clicked.
	The first three instructions set up the program to make sure the screen is clear, the sprite is the right size and the pen is 'up'.
	The next four instructions display messages to the user as if the sprite was speaking. The time the messages are on the screen, is in the instruction.
	Now we wait for the user to click the mouse. The program will not go further until the mouse button is pressed.
	A message is displayed and then we change ROTFL Cat into a pencil.
	A message is displayed and the MotorCyclePassing sound is played.
	Now we shrink Sprite1 to a more pencil-like size, . . .
	delay a little (1 second) so that things don't happen too quickly for the user and then . . .
	use 3 seconds to move the pencil onto the whiteboard picture. (The coordinates you get before writing the program, by moving the mouse on the stage to the position you want, and seeing what the coordinates are.)
	We are ready to draw the triangle. Before we start we must put the pen down,
	then we display more messages and start to play another sound to keep it interesting.
	Now we use multiple commands to draw the triangle.
	Pick up the pen and . . .
	finally, we finish with a triumphant pop sound.
	Display a message, change back into ROTFL Cat and . . .
	move back to where we started.
	There we grow back to full size and . . .
	display a final message.

Remark:

- The actual working part of the program simply draws a triangle – the rest is all to get the user involved and make the program user friendly.
- This (the drawing of a triangle) is the part that you would have had to use an algorithm to plan for.

ROTFL Cat says:

It is NOT unusual for the actual working, problem solving part of a program to be much, much shorter than all the rest that deals with communicating with the user and making the program friendly and appealing.

Most of the instructions in Scratch are as simple as you see them here – they are not difficult to understand and use. The real test / challenge in programming is to take a whole lot of very simple instructions and use them to create something both powerful and easy to use. That is where you must use a combination of creativity and hard work. Take the time and effort to add the extra instructions that make your program interesting and exciting, and you will end up with a pleasing product.

4 Activity • Some questions about the program

1. Would you describe the program as concrete or abstract? (Refer to the module 'Thinking for computers' for definitions of the words abstract and concrete.)
2. Explain the reasons for your answer in 1.
3. Part of the problem was working out how to draw a simple equilateral triangle (all the sides are the same length). So basically you start at a point, move a set distance, turn 120 degrees and so on... Now create a flowchart for drawing an equilateral triangle and add it to your algorithms file.
4. (Advanced work). Look carefully at the program.
   a) Write down the groups of instructions that are repeated (i.e. used in exactly the same format more than once).
   Identifying this kind of repetition is important because we can program more efficiently using repetition structures in our program (you will learn more about this later).
   b) You have already seen the use of repetition in algorithms. Draw a flowchart for drawing a triangle but use repetition rather than multiple repeated statements.
   Hint: You will need a counter variable and a decision.
5. What is the difference between the **[move]** and the **[glide]** program blocks?
6. What will happen if you remove the **[pen up]** program block?

ROTFL Cat says:

Follow the link **Print a program** on the **Scratch is gr8! Help file** to see how you can create a neat printout of a program. Your teacher will give instructions on the programs you should print and file in your algorithms file.

5 Activity • Draw your own shapes

1. What would it take to draw a hexagon? Change the program so that it draws a hexagon instead of a triangle. Use the **Save As** option to save the program with a new name, then only change the relevant program blocks.

2. Open the program **Draw a yellow line**. Write a script using the following pseudocode:

```
Begin
    Point to the right (90 degrees)
    Go to position (-195, -92)
    Remove all drawings
    Put the pen down
    Change the pen's colour to yellow
    Set pen's size to 10
    Move 100 steps
    Turn 45 degrees anti clockwise
    Move 100 steps
    Turn 45 degrees clockwise
    Move 100 steps
    Turn 90 degrees anti clockwise
    Move 100 steps
    Turn 90 degrees clockwise
    Move 100 steps
End
```

Hint: This program executes so fast, you cannot see how the figure is drawn. You can slow down the execution of the program. Click on **Edit**, **Set Single Stepping**, **Flash Blocks (fast)**; **Edit**, **Start Single Stepping**.

Run the program again. To make it go at 'normal' speed, click on **Edit**, **Stop Single Stepping**.

You can also add a **[set pen color to]** block before every **[move 100]** block to change the colour for every line. This will help you to analyse the program.

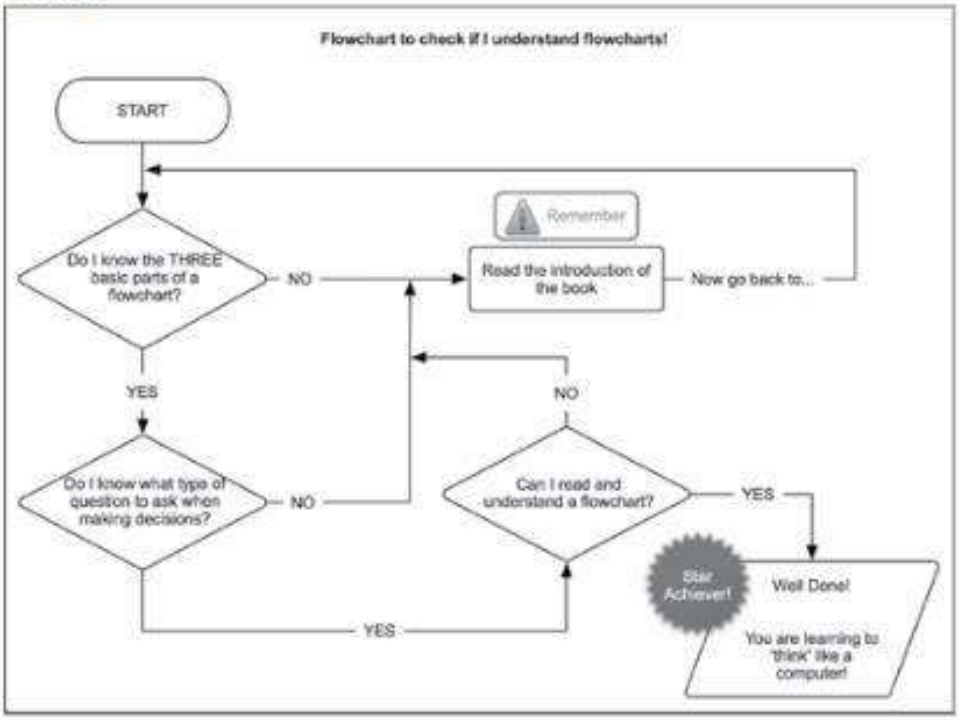
Teach a sprite to repeat actions

One of the most important skills you need to practise and develop, is the ability to see where repetition takes place in the steps you take to solve a problem. (In question 4 of activity 4 you were asked to identify repetition.)

Repetition in flowcharts

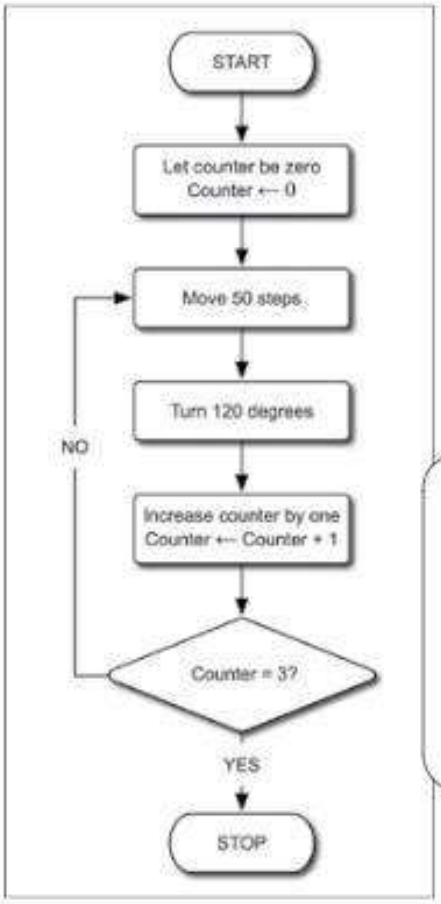
Although we have not discussed it formally, you have already seen many examples of repetition in algorithms. It is usually quite simple to represent. Do you remember this flowchart?

The flowchart contains three repetition structures. Can you figure out what they are? (If you don't remember exactly how flowcharts work, read the introduction of the book.)



Repetition in a flowchart is as simple as drawing a line from one block so it points to another block which has been executed previously. What is important is:

- The repeated steps must end with a decision that will either take you back to the start of the repetition process, or will let you exit the repetition.



- During the repetition process something related to the ending question must be given the opportunity to change. In computer programming this normally means changing the value of a variable.
- A fixed number of repeat cycles normally use a 'counter' – i.e. a variable that has 1 added to it every time the loop repeats. Eventually the counter value matches the number of times the repetition must happen and so you move on to the rest of the program.

Based on the above, a flowchart for drawing a triangle that includes repetition will look like this:

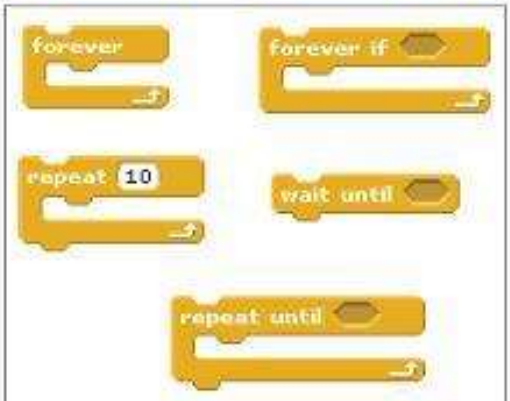
When looking for repetition you need to check:

- When does the repetition start?
- What instructions are repeated?
- What does it take to stop the repetition?
- This part is very important. Does the repetition happen a fixed number of times – like the three times the same instructions are repeated to draw a triangle, or does the repetition keep on happening until some condition has changed – like waiting for the user to click the mouse?

6 Activity • Flowcharts with repetition

Based on the example above, create flowcharts for the following:

1. Drawing a square.
2. Controlling a traffic light (green, orange, red).
3. Drawing a house using repetition to draw the building (square), roof (triangle), doors and windows (rectangles).



The different repetition structures in Scratch.

Repetition structures in Scratch

Why is identifying repetition important?

- There are special structures in all programming languages that make working with repetition easy.
- Using repetition instead of writing out the same instructions many times over reduces mistakes.
- Using repetition makes it easier to change and update your program.
- Using repetition makes it easier to find and fix 'bugs' (problems) in your program.

Scratch provides you with no less than five different ways to do repetition in your programs (They are all found in the **Control** instruction set).

Only one of the repetition structures deals with repeating steps a fixed number of times. The other four all deal with checking to see if something has changed to decide if the repetition must stop or not.

Before you even think about using repetition in your program, you must think about how the repetition will end.

The rest is simple: the instructions inside the repetition structure will repeat until the condition to end the repetition is met!

7 Activity • Analysing repetition structures

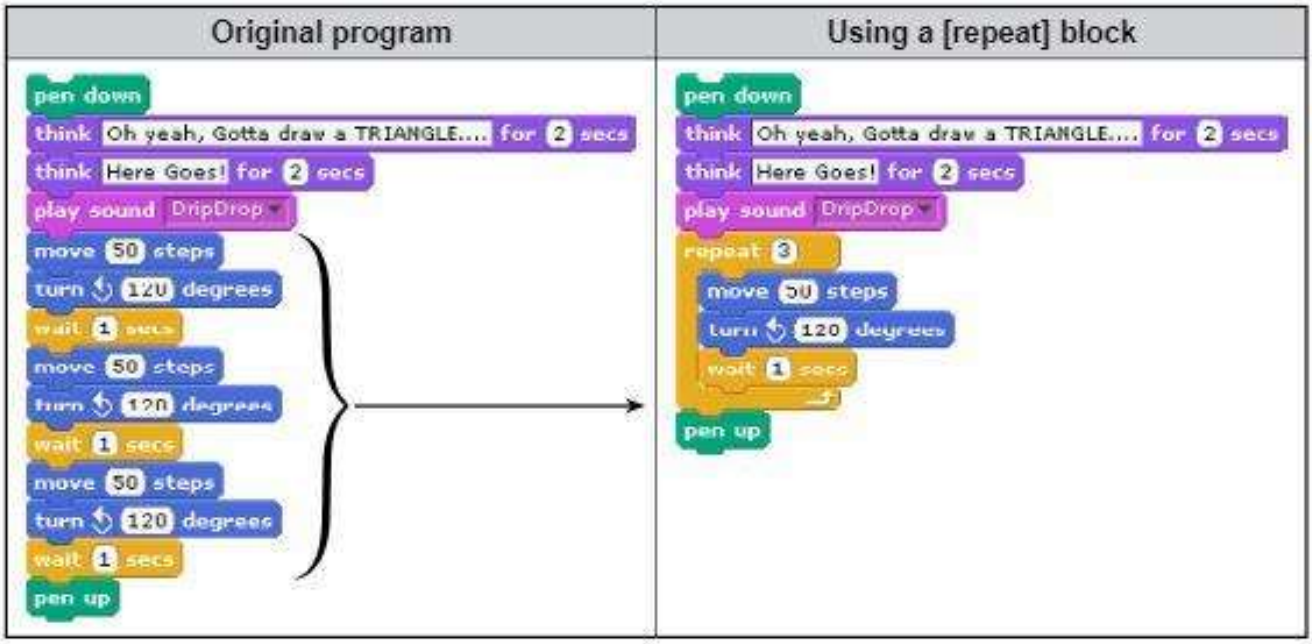
Take a look at the repetition structures that Scratch offers you. Think about them and then answer the following questions:

1. Can you tell which one will repeat only a fixed number of times?
2. Which structure will never stop repeating?
3. Which structure contains no instructions?
4. Which structures will repeat their instructions at least once?
5. Which structures might never repeat their instructions at all?

Repeat program blocks

In the flowcharts we constructed, we had to manually increase a counter variable and test if it has reached the end value to create a repetition structure. Luckily Scratch has a structure available where you only need to specify the number of times a group of instructions need to repeat, and the programming language updates and tests the counter automatically. We are going to look at the use of this structure now.

We'll start off by looking at what happens to the triangle drawing program if we re-do it with a repetition structure. We only include the statements between **[pen up]** and **[pen down]**, the rest of the program stays exactly the same.



Before the change could be made, the question of how the repetition should end had to be asked and answered. Because a triangle only has three sides, and all we needed to do was repeat the process of drawing a line three times to draw the triangle, it was fairly easy to come to the decision that the repetition would happen a fixed number of times.

This means that **[repeat (10)]** is the obvious repetition structure to use. Simply click on the 10 and change it to a 3 and you have a repetition structure that will carry out the instructions inside it three times. Now all you have to do is drag the instructions that must be repeated inside the structure.

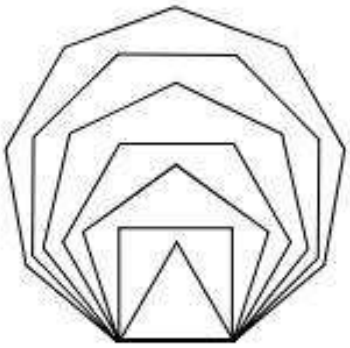
All Scratch's repetition structures work the same way – the only difference is how it decides if the repetition will end.

ROTFL Cat says:

Follow the **Modify scripts** link on the **Scratch is gr8! Help file** for useful tips on modifying scripts.

8 Activity • Apply your knowledge to solve problems

1. What would have to be changed in the triangle-drawing program to make the triangle bigger or smaller?
2. What would have to be changed to make the program draw a
   a) square?
   b) hexagon?
   c) a five-point star? *(Hint:* You have to turn the sprite 144° after each line.)
3. Write the Scratch programs to draw the square, hexagon and five-point star using repetition.
4. Why is there a significant relationship between the repetition counter and the angle?
5. What would happen if we changed the repetition to 180, the distance to 1 and the angle to 2? Try it and find out! (You might want to take out the **[wait 1 second]** or else the program will run for 180 seconds!)
6. What would the repetition count need to be if you changed the angle to 4 and wanted to have a similar result as with the program in question 5?
7. Challenge: Recreate this shape.
   Hint: The first shape has 3 sides, the second one 4 sides and so it continues until the last shape which has 9 sides.



Create animation

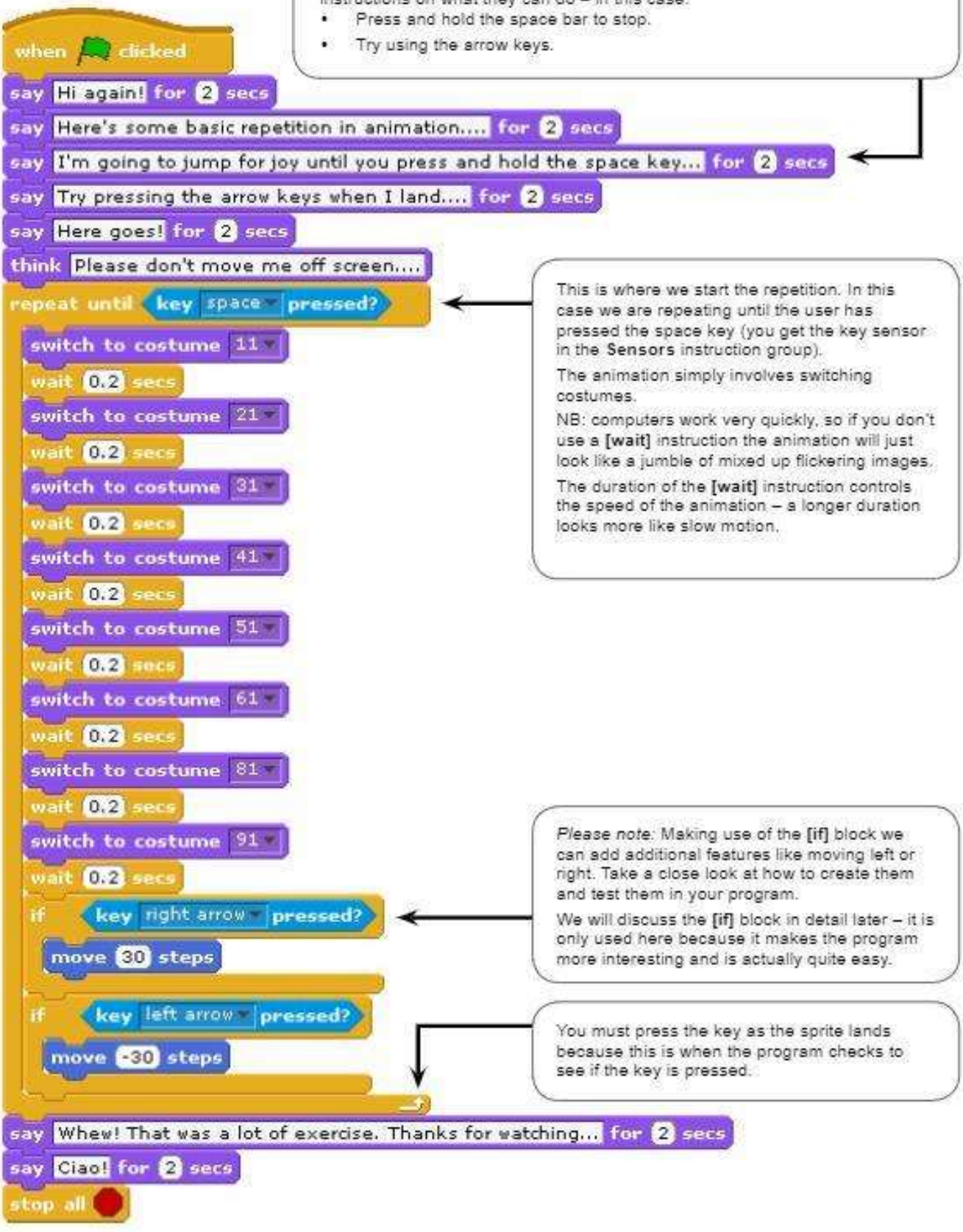
Use repetition to create animation

In the triangle example we have already used basic animation. We saw how we can use the **[glide]** instruction to move a sprite – but the problem is that this just moves in a straight line. If we use a repetition structure combined with **[move]** commands and/or even costume changes, we can create interesting animation.

The example which follows will show you how it is done.

9 ACtivity • Use repetition and Costumes to create animation

Create the following program:



It is always best to tell the users what is going to happen – and give them clear instructions on what they can do – in this case:

- Press and hold the space bar to stop.
- Try using the arrow keys.

This is where we start the repetition. In this case we are repeating until the user has pressed the space key (you get the key sensor in the **Sensors** instruction group).

The animation simply involves switching costumes.

NB: computers work very quickly, so if you don't use a **[wait]** instruction the animation will just look like a jumble of mixed up flickering images.

The duration of the **[wait]** instruction controls the speed of the animation – a longer duration looks more like slow motion.

Please note: Making use of the **[if]** block we can add additional features like moving left or right. Take a close look at how to create them and test them in your program.

We will discuss the **[if]** block in detail later – it is only used here because it makes the program more interesting and is actually quite easy.

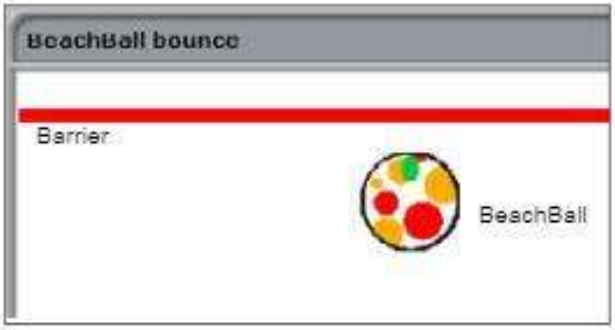
You must press the key as the sprite lands because this is when the program checks to see if the key is pressed.

Repetition is a very useful tool in programming. So, whenever you are faced with a problem where you have to develop an algorithm and write a program, keep a sharp eye out for where you can use repetition structures. Here are examples of repeat structures with stop conditions you have already used or you may want to use.

Animation Techniques

In the following examples we are going to show you a few techniques you can use in your own programs to create movement (animation) and special effects. We are going to see how

- a program can contain more than one sprite
- we can change the direction the sprite moves in



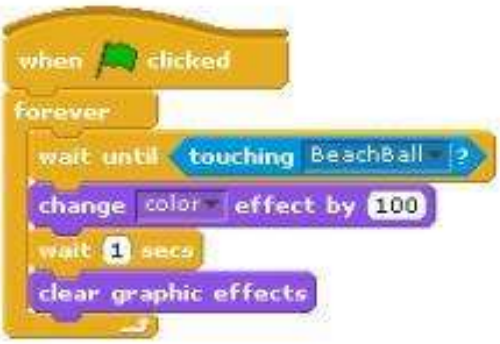
- colour effects can be used to make a program more spectacular
- the computer can decide how fast a sprite should move.

This program has two sprites – a beachball and a barrier. The barrier is a red rectangle created using the Editor.

Script for the BeachBall

When the flag is clicked, the ball moves to the centre of the stage and turns 'up' (to the top of the stage). It keeps on testing if the barrier has been touched – if not, it moves 10 steps (a positive value so it moves to the top of the stage). As soon as the barrier is touched, the ball starts moving 10 steps at a time to the bottom of the stage. (The number of steps is now a negative value.)

Script for the Barrier



When the flag is clicked, the barrier keeps on checking if the beachball is touching it. As soon as the ball touches it, the colour changes, and after 1 second it turns red again.

Direction of a sprite

0: The sprite will face in the direction of the top of the stage – 'up'.

180: The sprite will face in the direction of the bottom of the stage – 'down'.

90: The sprite will face in the direction of the right-hand side of the stage – 'right'.

-90: The sprite will face in the direction of the left-hand side of the stage – 'left'.

10 ACtivity • Move minus

Open the program **BeachBall bounce**. Execute the script marked **Script 1**. When you open the program, you will see a green dot on one side of the beachball. In this script the green dot always stays in the same spot – the ball just moves up and down the stage.

In Scratch it is easy to follow the blocks as they execute visually.

See how it works:

- Click on **Edit**, **Set Single Stepping ...**, **Flash blocks (fast)**.
- Execute the program.
- Every step which executes will flash yellow. While the program executes, you can click on
- the other sprite and see how the scripts execute at the same time.
- Click on **Edit**, **Stop Single Stepping** to return to normal execution.

ROTFL Cat says:

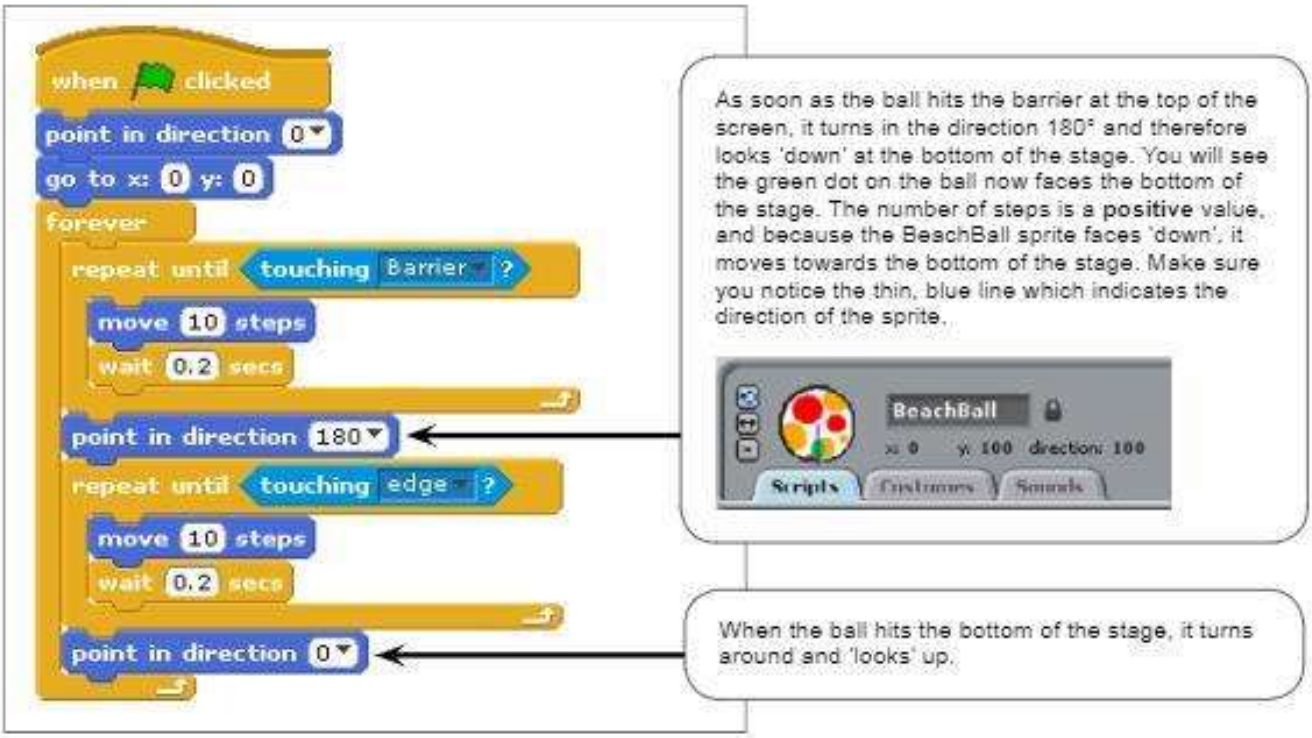
The scripts for different sprites can execute at the same time. It is called **hyperthreading.**

In computer programming there are often many different ways in which results can be obtained.

11 ACtivity • Change direction

Activate the script marked Script 2 in the program **BeachBall bounce**. (Separate the Hat Block of Script 1 from the rest of the stack, and snap the Hat Block of Script 2 onto the stack below it.) Execute the program. Explain what behaviour of the program is the same as when Script 1 was executed, and what is different.

The script for the BeachBall sprite in Script 2 is as follows:



The behaviour of the sprite is the same in that the BeachBall sprite still moves up and down the stage. The difference is that the direction of the ball changes so it faces 'down', and therefore it moves a positive number of steps.

In the previous scripts the BeachBall sprite always moved 10 steps at a time. However, especially when you create games, you want sprites to be able to move at random distances. Scratch has a function which can provide random values – so in effect the computer can control the speed at which the ball moves.

12 ACTivity • Fast / slow

Activate the script marked **Script 3** in the program **BeachBall bounce**. (Remember to separate the Hat Block for Script 2 from the rest of the stack.)

Program blocks when the BeachBall sprite moves 'up'

Here any number between 1 and 20 (both included) will be returned by the random function.

Program blocks when the BeachBall sprite moves 'down'

Here any number between 1 and 10 (both included) will be returned by the random function.

When does the ball move faster - when it moves up or when it moves down?

ROTFL Cat says:

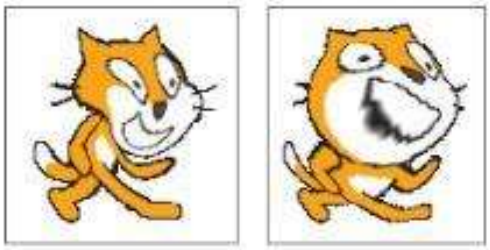
The random function can also return a negative value.

Hint to test the way a program block works:

You do not always have to write a complete program if you just want to test the way program blocks work quickly. You can place one or more single program blocks in a script. When you click on it, you will see the effect thereof.

Create a new project. Place the following program blocks in the Scripts Area, and experiment. Remember, there is no book which can teach you everything about a programming language. You need to be adventurous and investigate the different possibilities of the different programming blocks.

Click on **[clear graphics effects]** to return the sprite to normal.



13 ACtivity • Play with animation

Create a program with animation which meets the following criteria:

- Contains more than one sprite.
- One or more of the sprites must move.
- One or more of the sprites must change direction.
- Make use of colour and animation effects.
- A sprite must be able to react on more than one event. (For example, let the sprite move the centre of the screen if the space bar is typed, and move around when the green flag is clicked.)

ROTFL Cat says:

Before you get the chance to apply your creativity at the end of this module, let us ensure you are able to apply all the knowledge and skills you have gained so far.

14 ACtivity • What are you able to do?

Answers to these questions can be found in the resources:

Knowledge questions

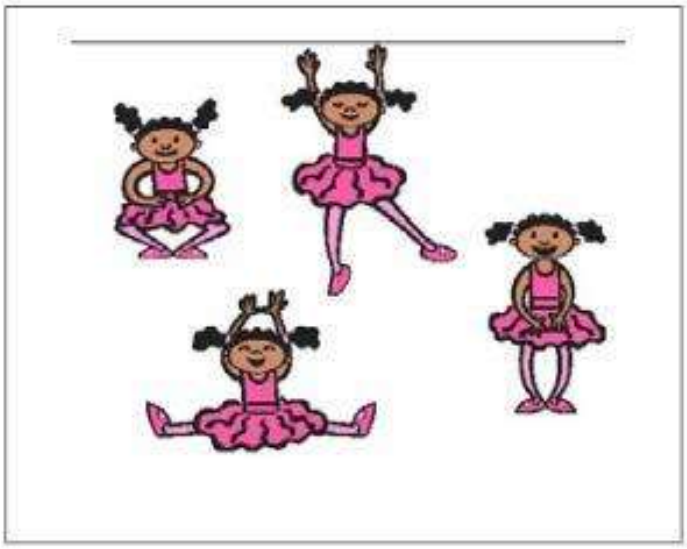
1. What are Scratch projects made up of?
2. How do you tell a sprite what to do?
3. What is the stage?
4. Where is the (0,0) coordinate of the stage?
5. What is the Sprite List?
6. From which area do you find the program blocks?
7. What type of files can be used for a sprite, a costume or a background?
8. What facility does Scratch offer to help you to follow the program blocks as they are executed?
9. What is the difference between a Hat Block and a Stack Block?
10. Name a few program blocks you can use to communicate with the user of the program.
11. Which program block can change the size of a sprite?
12. Which program blocks can cause a script to pause or be delayed?
13. Which program blocks can cause a sprite to move to another spot on the stage?
14. Which program blocks can cause a sprite to change direction on the stage?
15. Name a few events a sprite can react to.
16. Which repetition structure will cause the program blocks inside it to be executed repeatedly, and can only be stopped by clicking on the red dot?
17. Which repetition structures have the potential never to end (be an endless loop)? Explain why.
18. What does hyperthreading refer to?

Skills questions

19. How can you copy a stack of blocks to another sprite?
20. How can you write comments inside a script?
21. If you want to insert a block into an existing stack, what graphics will assist you to slot the block into the correct position?
22. How can you find out what a block does?
23. How can you change the name of a sprite?
24. How can you determine the coordinates of a position you want to move a sprite to?
25. In which ways can you change the costumes of a sprite?

15 Activity • Animation in Scratch – write your own programs

1. Create a program using a sprite with the 4 costumes displayed. Write a script to do the following:
   - Let the sprite move to the position (0, -100).
   - Let the sprite change position and costumes so the dancer seems to jump up into the air and down again.
   - Use repetition and decide how many times she should jump up and down.

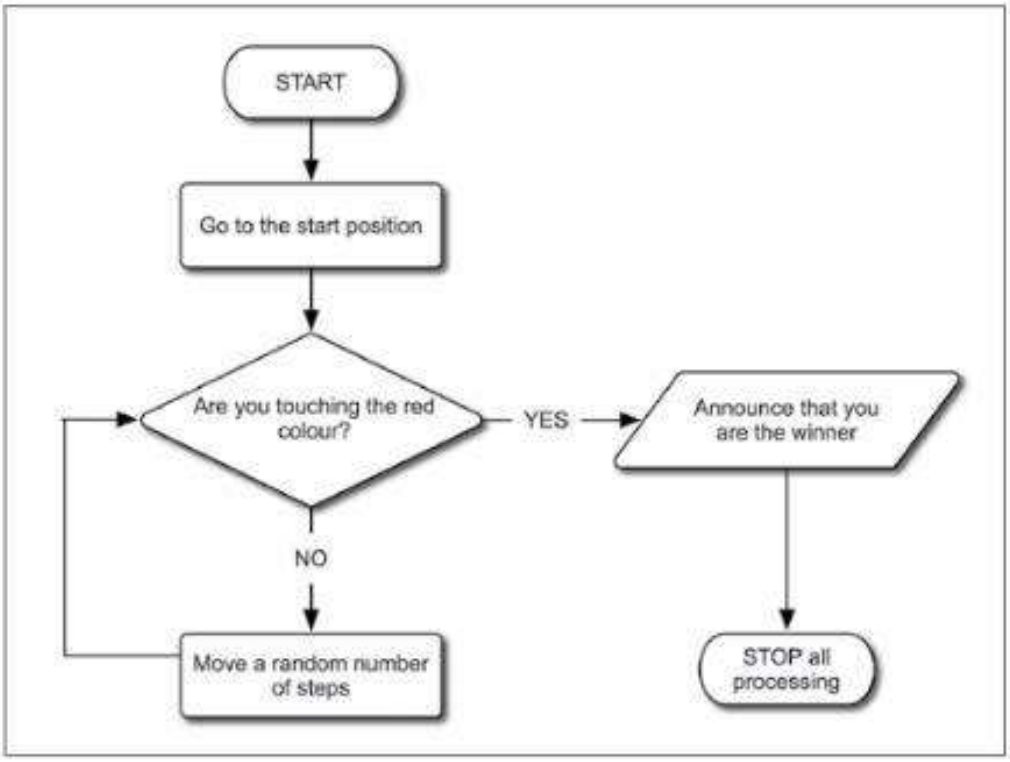


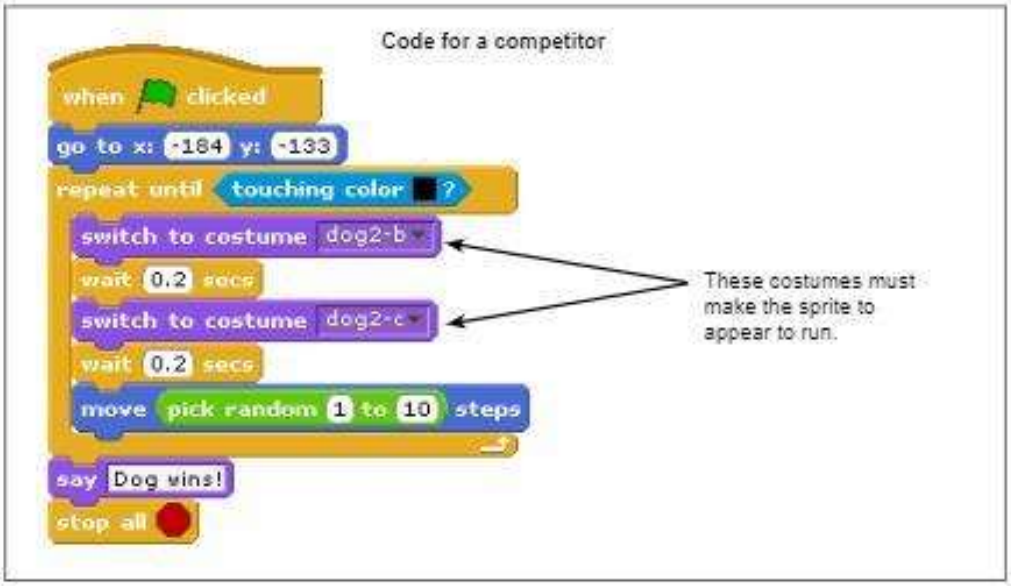
2. The Race

   You need to create a stage representing a race track with a start as well as a finish line. You also need two competitors (sprites). You are going to get your competitors to race across the stage. The first one to touch the finish line will be the winner.

   Set up a suitable background or use an existing stage. Use a sprite to create a finish line (just draw a thin rectangle).

   - We need a repetition structure to keep the race going for as long as it takes for someone to win!
   - We don't know how long it will take, so we can't use a loop with a fixed number of repetitions.... So we will use the **[repeat ..until]** block.
   - We need to check if the competitor has won – and luckily a sprite can see if it is touching a colour – so we can add a **<touching color>** condition to the **[repeat until]** block (you find this in the **Sensing** group). Remember to select the colour of the finish line!
   - We need to move the character inside the loop so we must use a **[move]** command – but a fixed distance means the competitor will always move at the same speed – and that means we don't have a real race so...
   - The key to turning it into a race, is to use random numbers for movement so that we don't know how far each competitor will move each time the instructions are repeated!
   - You find the random number generator in the **Operators** group.
   - So you will need a program that looks something like the instructions on the next page – and you need it for each competitor!
     Hint: Look at the explanation in the Shortcuts link in the **Scratch is gr8! Help** file to see how to duplicate code of one sprite for another one.





What will happen if . . . (first discuss this, then try it out to see what happens)

- you do not add the [go to] program block for each sprite?
- the [say .. for 2 secs] program block have been used instead of the [say]?
- you do not add the [stop all] program block?

3. Create an animated story of your own choosing. This should include at least (but not only):
   - Movement
   - Sound
   - Narrative (words displayed by talking and thinking characters)
   - Limited user interaction (like pressing a key or clicking a mouse)
   - Changing costumes
   - More than a plain white stage
   - Some experimentation and enthusiasm and creativity on your part!

MODULE 1.3

Input, processing, output

Introduction

In Module 1.2 you learnt how to use the interface of Scratch and how to create animation using repetition. In this module you are going to learn how to write programs for algorithms such as those you designed in Module 1.1. You will use variables, obtain input from the user, create programming statements to do processing and to produce output – the results.

Teach a sprite to 'remember' by using variables

Creating variables

Every programming language allows you (the programmer) to create and use variables. However, when you are writing a program, you have to be a bit more specific as far as variables are concerned than when creating an algorithm. For example, you cannot simply tell a computer to 'Store the value 25 in the variable "Number"'.

- For a start, the computer doesn't know what 'Number' is.
- Next, it doesn't know how much space in memory should be used to store the data.
- It also doesn't know how to treat the data – as a number?, as letters and words?

This is why most programming languages require you to do the following to take care of these problems:

- A variable must be declared (given a name).
- The type of data to be stored in the variable must be indicated.

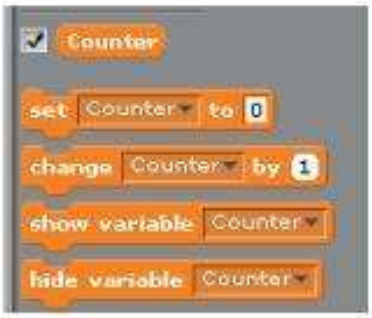
Creating a variable in Scratch is easy. Follow these steps:

ROTFL Cat says:

When looking at the declaration of variables, Scratch is not the same as most other programming languages. It does not require you to specify the type of a variable. If you use a variable to do arithmetic, Scratch will automatically assume the variable is a number.

1. Click on the **Variables** instruction group.
2. Click on the **[Make a variable]** button – a dialog box pops up.
3. Give the variable a name. In this case, call it **Counter**.
4. Choose whether the variable should be used only for/by this sprite, or if it should be available to all sprites. (More about this later.)
5. Click **OK**.

You can determine whether your variable was created successfully by looking at the **Variables** instruction group again. Additional instructions will be available in the group.

New instructions available	What they do
	The first block represents the variable, Counter – you can drag the variable into blocks such as the mathematical operator blocks, where it will be used in calculations or instructions. **[set]** assigns a value to the variable. **[change . . by]** allows you to add a specific value to the variable (if you make the value negative, then the value will be subtracted from the variable). **[show]** and **[hide]** either shows or hides the variable value from displaying on the stage. Showing can be useful whilst we are writing and testing our program, but we usually want to hide the variable when the program is running.

A variable name such as Counter suggests that a number value will be stored in it. This can be used in any mathematical operation, and also in place of numbers in program blocks. For example, instead of using **[repeat 3]**, you can use **[repeat Counter]**, and the repetition will happen as many times as the value stored in the Counter variable.

Variable types in Scratch

Variables in Scratch can be based on one of two structures:

- **Variable**
  A data structure which can contain a single value only. This can be a ***number*** or a ***string*** (a sequence of characters) or a ***Boolean*** (True/False) value.
- **List**
  A collection of variables under one name – the variables are stored in numbered boxes. We will learn about lists a little later.

The difference between the number, string and Boolean types is what can be done with them in programs.

Number	• Stores numbers. • Can be used for calculations. • Can be used wherever number values would be used in other Scratch instructions.
String	• Stores a sequence of any characters. • Cannot be used for calculations. • *Please note:* Scratch will not crash and give an error message when you put a variable that holds a string into a place where a number value is expected. Instead it will simply ignore the string and use a value of 0 as the expected number when the instruction is executed.
Boolean	• Stores a True or False value. • Cannot be used for calculations. • Can store the results of a comparison. • Can be used for making decisions.

ROTFL Cat says:

Scratch does not expect you to tell it whether the variable you create will hold letters or numbers or Boolean values. It makes intelligent decisions about the type of data in a variable on its own.

Most other programming languages want to know what type of data a variable will hold. To get used to this way of thinking, it is best that you

- decide for yourself what type of data the variable will be used for when you create the variable (string or number or Boolean value) and give it a descriptive variable name.
- make sure that you only use it for that type of data.

Naming variables

Scratch does not really have rules about variable names. This is very unusual, as most programming languages have strict rules about naming variables. To prevent you from getting all confused later when you move to a different programming language, we suggest that you apply the following rules when naming variables:

- Start your name with a letter, not a number or symbol.
- Do not use spaces.
- Do not use other symbols besides – and _ (minus and underscore).
- The name should describe the content of the variable.
- Use a capital letter at the start of every word – e.g. CellNumber, DateOfBirth.

Scratch will not give you any errors if you do not follow these naming rules, but most other programming languages do apply these rules.

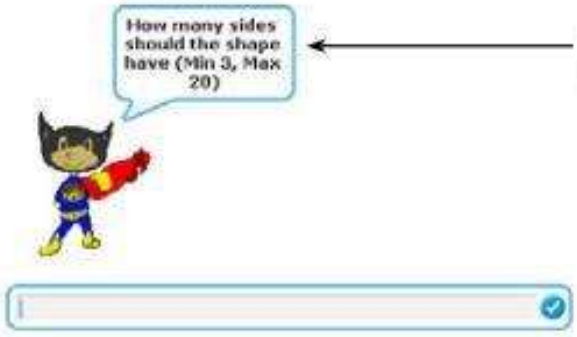
Input from the user

The last thing we need to do before we actually start to work with variables, is learn how to store values entered by the user into variables.

Consider the following two blocks which are found in the **Sensing** instruction group:

[ask]: Displays a speech bubble with the question you set and then waits for the user to type in a value and either press the Enter key or click on the ✔.

(answer) is a built-in variable in Scratch. It holds whatever the user has entered the last time that the **[ask]** block was used.



This is an example of the display of an **[ask]** block on the screen.

Let us write a program to see how the user can enter an integer value that will then be used to control the number of times an action will take place on the screen.

Activity • Use a variable in a [repeat] block

1. Start a new project.
2. Load the two dragon costumes for Sprite1 as indicated in the screen shot that follows (delete the default costumes).
3. Create a new variable called NumberCoughs (for this sprite only).
4. Create the script for Sprite1 using the program blocks displayed.
5. Run the program.

Consider the program that you have created in Activity 1, and note the following:

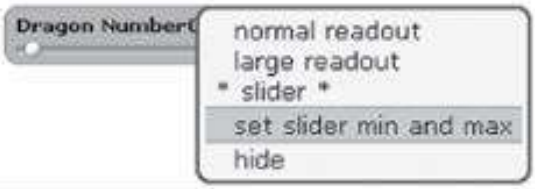
- Once a user has entered the integer value in the **[ask]** input dialog, Scratch will automatically place the value entered in a variable called (answer). (Answer is Scratch's own variable.)
- You have to transfer the value stored in the variable (answer) to the variable you created yourself.

This is similar to Mathematical statements such as: x = y, or in this case:

NumberCoughs = answer

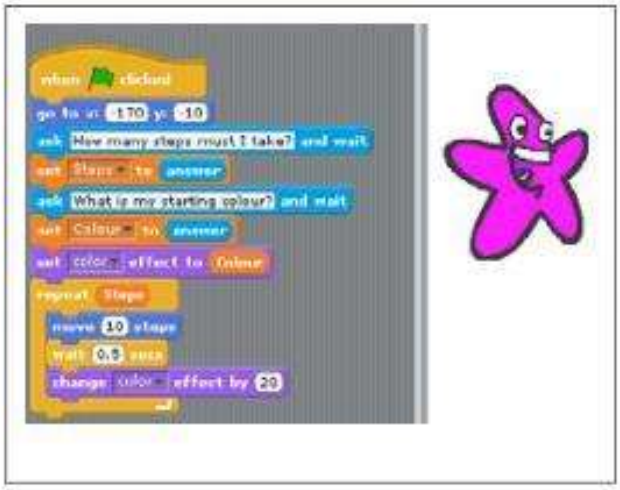
- You can choose to have the value of the variable NumberCoughs displayed on the stage or not. This can be done by ticking the check box next to the variable name in the **Variables** instruction group. You can also use the program block **[Hide variable NumberCoughs]** or **[Show variable (NumberCoughs)]**.

The small window on the stage displaying the value of a variable is called a monitor. There are three different formats in which the monitor can display a value. (Right-click on the monitor to choose the display you prefer.)

	A small window displaying the name of the sprite, the name of the variable as well as the value stored in the variable.
	A larger screen with only the value stored in the variable.
	A slider with which you can change the value of the variable. You can also set the minimum and maximum value that the slider should allow. 

2 Activity • Use more values provided by the user

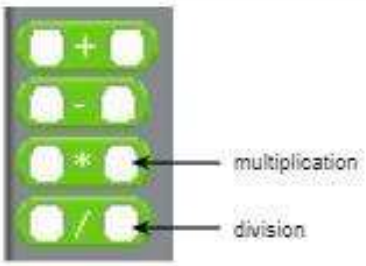
1. The following two scripts make use of input from the user to control what should happen. Create the two programs, and experiment with the values in the various program blocks. In this way you will learn more about the use of colour and sound in programs. (Note that the colour chosen should be typed as an integer value.)



2. Open the program called **Fan Spin**. Change the value of the variable FanSpeed using the slider and see the effect.

Teach a sprite to do calculations with numbers

The use of variables (abstraction) is not really useful without a little bit of manipulation. Consider the four program blocks (in the **Operators** instruction group) used to do basic arithmetic operations.



You can put a fixed value (a number such as 23) or a variable in either of the white boxes and Scratch will perform the calculation.

Here is an example of the way in which Scratch can do calculations and present the answer.

Order of precedence in calculations

We have been taught how to do calculations and apply mathematical rules. For instance, if we have to calculate 3 + 4 x 5 = 23, we know that we have to do the calculation (4 x 5) first, and then add 3. Programming languages do not always have the knowledge of the order of precedence for mathematical calculations built in. As programmers we then have to tell them which calculations to do first.

If you look at the available program blocks, you will see that it is not possible to type brackets in a mathematical operators block, e.g. 3 + (4 x 5). The effect of brackets is achieved in Scratch by making use of **stacking**.

3 Activity • Use stacking to indicate the order of precedence in calculations

Create a small program containing **[say]** program blocks to calculate 3 + (4 x 5).

Step one: Drag the addition block to the Scripts Area first.

Step two: Plug the multiplication block into the open box on the right-hand side of the addition block.

Notice how the  appears to sit on top of the other block.

Scratch will perform the calculations at the top of the stack (therefore the *) first.

Test it by entering numbers.

Tip: You do not need to write a complete program using a sprite and program blocks. You can use a mathematical operation block only, enter values and double-click on the block. A speech bubble will appear containing the answer

4 Activity • Let Scratch do the work of a pocket calculator

1. You scored 45 out of 60 for a test. Use mathematical operation blocks to calculate your percentage.
2. Calculate the average of 68; 89 and 77.
3. Calculate (3 + 2) / (1 + 1)
4. Determine the velocity of a car that travels 14 km in 8 minutes in km/h.

When a sprite needs to produce output, it is not very useful just to let a number appear on the screen. You at least want to see a descriptive message indicating what the value means, e.g: 'The average is: 10'. To do this the **[join]** operation block (in the **Operators** instruction group) is used.

Use number variables in calculations

If someone does not know how to calculate a percentage from his/her test marks, you can write a program using variables to do the calculation.

Here the **[join]** operation block is used to join a sentence and a value. You will need to do this in the activities which follow.

Note:

Since you are not going to use the answer for any other calculations in the program, you do not need to store the result in a variable – you only need to display the answer. However, if you want to display another message showing the increase in pocket money based on the percentage the learner scored, the program might look like this:

5 Activity • Explore working with variables

For each of the programs you need to develop a flowchart or pseudocode using variables to indicate how the calculations will be done. Develop the program only after having developed the flowchart or pseudocode.

1. Create a program where the sprite asks the user for the current year and the user's year of birth, and then tells the user how old he or she will be in the current year.
2. Calculate the amount each person must pay as part of a group bill tip at a restaurant (tips are usually calculated as 10% of the bill).
3. Work out the fuel consumption for a car in litre / 100 km if you are given the amount of fuel used and the distance travelled.
4. Determine the amount of money in Pounds that a person can buy. You are given the exchange rate, as well as the amount, in Rand, that he or she wants to spend.
5. Determine the amount of money in Rand that a person can buy. You are given the exchange rate, as well as the amount, in Pounds, that he or she wants to spend.
6. Ask the user to enter the temperature in Fahrenheit. Convert it into degrees Celsius by applying the formula Celsius = (Fahrenheit – 32) x 5/9.
7. Write the program for the pseudocode provided.

```
BEGIN
    Ask 'What is price 1?'
    Obtain Price1
    Ask 'What is price 2?'
    Obtain Price2
    Ask 'What is price 3?'
    Obtain Price3
    TotalPrice = Price1 + Price2 + Price3
    VAT = TotalPrice x 0.14
    FinalPrice = TotalPrice + VAT
    Display 'Subtotal: R', TotalPrice
    Display 'Vat: R', VAT
    Display 'Total: R', FinalPrice
END
```

8. A person lays floor tiles at a tariff of R30.45 per m². Calculate the total cost to tile a room of a certain length and width (as specified by the user). The person wants a deposit of 10% of the cost prior to commencing, and the balance on completion of the work. Calculate the deposit as well as the amount that has to be paid on completion of the work.

Using mathematical functions

Scratch can do more than just the basic arithmetic that you have already encountered. It has a few mathematical functions that you can use. Let's take a quick look at what these functions are, and what they can do.

Round rounds a number to its closest integer – any decimal from 0.5 and up rounds to the next bigger whole number, and any decimal less than 0.5 rounds down to the integer. For example: Round (89.3) = 89; Round (63.49) = 63; Round (12.7) = 13; Round (109.8) = 110.

Mod does division, but gives the remainder of the division as a whole number – for example.: 10 mod 3 = 1; 10 mod 2 = 0 (there is no remainder); 10 mod 8 = 2; 125 mod 60 = 5.

Abs gives the absolute value of a number (for example if it is negative it removes the minus sign).

Sqrt gives the square root of the number – the answer will always be a decimal (real) number.

Sin, ***Cos***, ***Tan***, ***Asin***, ***Acos***, ***Atan*** are Trigonometric functions useful for drawing complex geometric shapes.

Ln gives the natural logarithm of any number.

Log gives the decimal logarithm of the number supplied. For example: Log (10) = 1.0; Log(100) = 2.0.

e^ gives you the decimal number of the natural logarithm.

10^ raises 10 to the power of the number supplied – for example: 10^2 = 100.

You may use these functions in any combination with each other and with the standard arithmetic operators.

6 Activity • Use mathematical functions

1. A group of friends go to a restaurant. They only have a certain amount to spend. They know the final amount to pay will be calculated by adding VAT of 14% to the amount they spend on food and drinks. They will also have to pay a tip of 10% on the amount with VAT included.

   Write a script so that a sprite will ask how much money they have available, and then work out (rounded to the nearest Rand) how much they will be able to spend on food and drinks.

2. A carpenter has to build a number of right-angled triangles made of wood. The company he works for provides the lengths of the two shorter sides of the triangle. He then needs to calculate the hypotenuse using the Theorem of Pythagoras (hypotenuse ^{2} = side1^2 + side2^2).

   Write a script so that a sprite will ask the lengths of the two shorter sides, and calculate and display the length of the hypotenuse.

Integer and real division

As explained before, Scratch does not require a programmer to declare the type of a variable. The type of a variable is determined by the way in which it is used. Look at the following table with examples of operations and resulting data types.

Operation	The variable Result	
	Value	Data type
	5	Integer
	3.3333333	Real
	2.46	Real
	60	Integer
	36 (Round will always provide an integer value.)	Integer
	5.0 (The sqrt function will always provide a real value – even if the resulting value has no decimal part.)	Real
	3.1622776	Real

One type of calculation that Scratch does not offer, is that of so-called 'integer division'. In some calculations we would like to disregard the decimal part of the result of a division operation. Look at the following example:

If I have 20 sweets, and 3 children each need to receive the same number of sweets, you will immediately be able to say that each child should receive 6 sweets. You calculated it as follows:

NumEachChild = 20/3 = 6.66666

However, because you know that you cannot split a sweet, you disregarded the decimal part. Scratch will be able to calculate how many sweets will remain after each child received 6 sweets – you can use the following calculation:

To obtain the integer part of the result, you can perform the following calculation:

7 Activity • Use integer division and MOD in calculations

1. The timekeeper of a marathon can only supply the time it took an athlete to complete the race in seconds. However, the athletes want to know the time in minutes and seconds. For example, 126 seconds is 2 minutes and 6 seconds.

   Write a script so a sprite can ask the athlete how many seconds he/she ran, and then supply the time in minutes and seconds.

2. A farmer sells eggs. She uses containers in which 8 eggs can be placed.
   Write a script to obtain the number of eggs she wants to sell, as well as the price for one container, and then display how many containers she will be able to fill, how many eggs will be left after filling the containers, and how much money she will receive after selling all the full containers.

3. A lawn is mowed at a rate of 2m² per minute. Create two variables (Length and Width) and display them on the stage using a slider. These variables must contain the length and width of the lawn. Calculate in minutes and seconds how long it will take to mow the lawn (e.g. 128 seconds is 2 minutes and 8 seconds).

Combine variables, animation and mathematics

When you combine repetition with variables, you can produce output with far fewer instructions!

Draw a regular shape

In a previous program we had to draw a triangle. But that was all the program could do. If you wanted to draw a square or hexagon, you would have needed to rewrite the program. If we think abstractly (in terms of variables) about what we do when we draw any regular shape, we end up with the following steps:

- To draw a shape, we start and end at the same point.
- This means we must turn through 360 degrees.
- The number of degrees we turn after drawing a side is constant, and is calculated by (360 / number of sides).
- So…

  Repeat for the number of sides the shape has

  - *Move distance*
  - *Turn (360 / number of sides)*

That is the algorithm for drawing any regular shape.

Drawing any regular shape (using repetition and a variable)

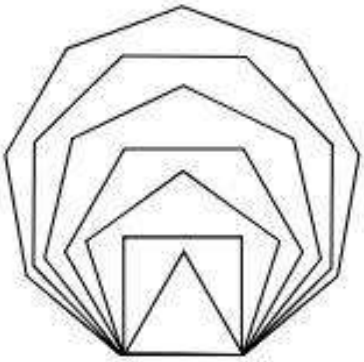
A variable called NrSides is used to store the value the user will enter.

To draw the shape we need to:
- add a repetition structure.
- use a **[move]** instruction, but instead of a fixed value, insert a calculation. This is because if we have a large number of sides, our shape would be too big to fit on the screen. So we make the length of the side shorter the more sides there are.
- insert the **[turn]** instruction using a calculation to work out the amount of degrees we need to turn.

Note: A large number of sides will look very much like a circle.

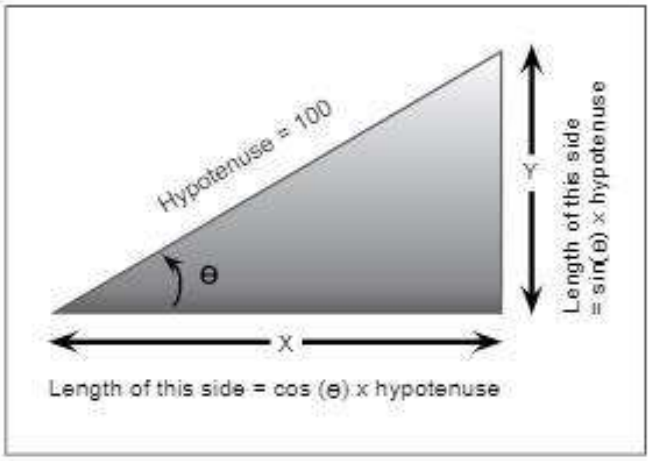
8 Activity • Apply your knowledge of variables, animation and mathematics

1. Modify the script you used in Module 1.2, Activity 7 to create this shape. Make use of variables.



2. Use the pseudocode that follows to write a program to draw a circle:

```
BEGIN
    Clear the screen
    Set the pen size to 5
    Pick up the pen
    Degrees = 0
    Repeat (90) times
        X = cos(Degrees) x 100
        Y = sin(Degrees) x 100
        Put the pen down
        Go to (X, Y)
        Degrees = Degrees + 4
END
```



Change the program to do the following:

a) Make use of a variable **Hypotenuse** which can be set using a slider. Make use of this variable to calculate the x and y coordinates.

b) Make use of a variable **StepsOf**, which can be set using a slider to control the **[repeat]** block and adjust the degrees.

Note: If you move the sprite to another position and then run the program again, the circle will not be displayed correctly. In Module 1.4 you will learn to use a program block which will solve these problems.

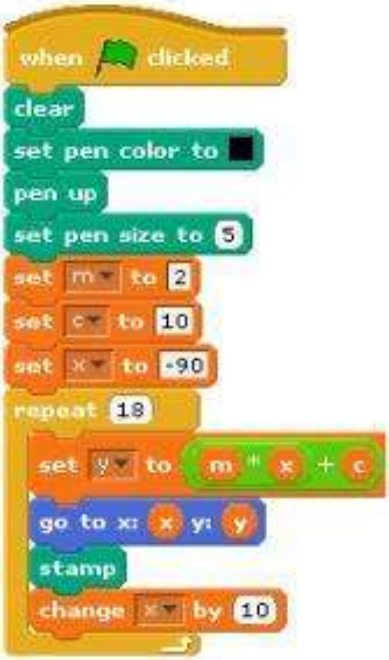
Draw graphs

Scratch can be used to draw graphs – it even has a background showing the X and Y axes (the background's name is **xy-grid**).

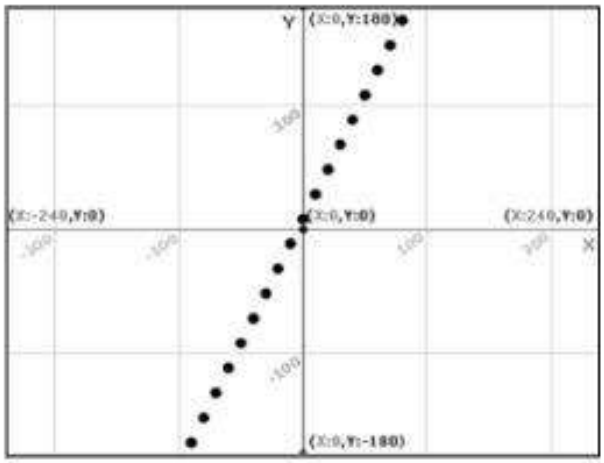
You should be familiar with the equation for a straight line: $y = mx + c$. The following program demonstrates how a script can be written to let Scratch draw a straight line on the stage using this formula.

9 Activity • Experiment with values to draw a line

- Open the program **straight line draw**. Change the values of m, c and x and note how the graph changes.
- Why must the following be true?: $-240 \leq x \leq 240$; $-180 \leq y \leq 180$



Scratch also has a program block that you can use to make copies of a sprite. You can use this in the line graph program to create a dot at each (x,y) coordinate instead of drawing a line. Replace the **[pen down]** block with the **[stamp]** block.



10 Activity • Draw using mathematics

1. The formula to create a parabola is $y = ax^2 + b$. Create a new program using the xy-grid as background, and draw a parabola.
2. Open the program **BullsEye**. Modify the program so it makes use of a loop.

Teach a sprite to handle strings

So far we have mainly looked at problems involving variables that hold *numbers*. All programming languages (including Scratch) are able to work with numbers as well as strings (sequences of letters, other characters and digits).

What is a string?

Simply put, a string is a sequence of characters. These characters include, amongst others, any character on the keyboard. Strings often need to be manipulated and changed – you might need to extract pieces of data from strings, change parts of strings, etc. All of this is called string handling or string manipulation.

To be able to work with and manipulate strings you need to understand how the contents of a string are stored in memory.

How strings are represented in variables

A string is represented in memory as a sequence of numbered bytes.

If we execute the given program blocks, and the user enters the name 'Peter', the data in the variable MyName can be represented as follows:

MyName				
P	e	t	e	r
[1]	[2]	[3]	[4]	[5]

We can refer to a specific character in a string by using the index of the character. The index refers to the position of the character in the string.

MyName[1] will therefore be equal to 'P', and MyName[2] will be equal to 'e', etc.

In some programming languages, each character can be addressed separately by placing the index in square brackets after the name of the variable. Scratch does not have this facility, but, as we will explain later, you will be able to extract one character at a time from a string in Scratch using the **[letter () of]** program block.

How the Computer sees strings

The computer stores characters as numbers. It uses a standard table of numbers associated with characters so that all computers everywhere in the world will associate the same number with its matching character. The table is called the ASCII table (ASCII stands for American Standard Code for Information Interchange).

Based on this we can see that the computer actually stores 'The' as 84 104 101, and you should also be able to work out that 'the' (116 104 101) is different from 'The' (84 104 101) because the ASCII table has different number values for 'T' (84) and 't' (116).

ROTFL Cat says:

There's a whole artform dedicated to creating art with letters and words. It is often called ASCII Art. Samples are provided. You can also search the Internet for ASCII art portraits of famous computer geeks like Bill Gates and Steve Jobs.

Scratch's string operators

In a previous activity you used the **[join]** operation block. This is not the only string operator that Scratch provides you with. All of the operators are shown and described below:

String Operators	What they do
 	**[join]** combines two inputs into one string. The inputs can be anything – a string, a number, a string or number variable or even a whole calculation. It is great for putting values together to create meaningful output for your program.
 	The **[letter () of]** operator needs a number in the first box and a string in the second, and it gives you the character at the specified position in the string.
 	The **[length of]** operator simply gives you a number that indicates how many characters are in the string in the white box.

Alone these string operators don't do much, but you can use them together with variables and other programming structures to create great programs!

You can also combine string input (from the user) and text that you want to display.

MyName is a variable you need to declare. You can type letters and/or numbers into the **[ask]** dialog. Since **[letter () of]** is a program block designed to work with strings, Scratch will automatically treat MyName as a string.

11 Activity • Display a Combination of your own message and variable values

Write a program to let the user enter three values: a name, a surname and then an age. An example of this program is given.

Note: You will have three variables (Name, Surname and Age), three **[ask]** blocks, but only one **[say]** block. You will have to use many **[join]** blocks in the **[say]** block to combine the variables and the other words that the sprite has to say.

12 Activity • Use string manipulation blocks

1. The program that follows creates a password by applying the following rules:
   - The password contains 6 characters.
   - The characters in odd-numbered positions can be any alphabet letter.
   - The characters in even-numbered positions will always be a vowel.

The program makes use of many variables.

a) Create the program, but see how few variables you can use to obtain the same result.

b) Change the program so it makes use of repetition to generate different passwords.

Hint: You need to set the variable NewWord to be an empty string. Thereafter you add single characters to the string and so 'build up' a new string.

2. When an amount is printed on a form, it often has the following format: R***23.67, so a person cannot add additional digits in front of the value.

   Create a program where the sprite will ask you to enter an amount (just the digits, not the R), and then produce a string containing a number of * between the R and the first digit. Assume the maximum value that can be entered, is 99999.99.

   Hint: You will need to use the **[length of]** program block.

13 Activity • Apply your knowledge of mathematical calculations, variables and string manipulation

Mathematical calculations

1. A book of 270 pages weighs 550 gram. Develop a Scratch program so you can enter the weight of any book in grams, and calculate and display how many pages the book will have. Assume that the weight of a single page in all books are the same.

2. Jason bought some eggs from a farmer at cost price per egg. Unfortunately, 5% of the eggs broke, but he sold all the other eggs at 25% more than cost price.

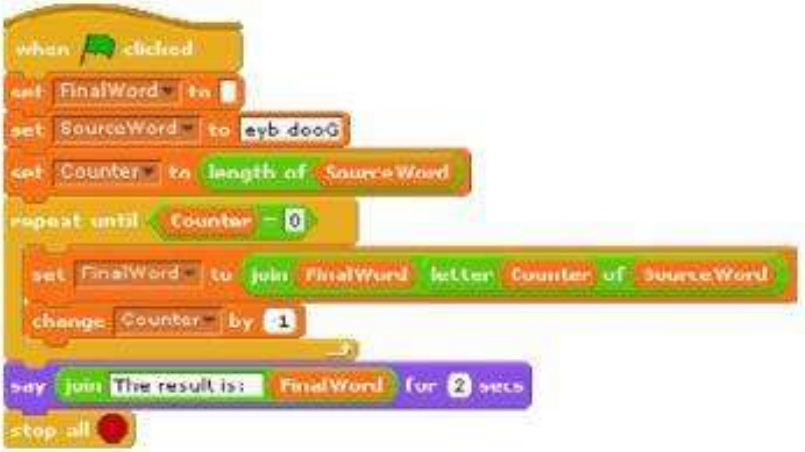
   Develop a Scratch program where the sprite will ask the user for the number of eggs Jason bought from the farmer, as well as the cost price per egg. The program should then calculate and display the profit.

3. The basic price for a phone call is 50c for the first three minutes. For every minute after the first 3 minutes (4 or more), an amount of 15c per minute will be charged. Tax of 14% must be added to the final amount (cost for the first 3 minutes + cost for additional minutes over 3 min) for a call.

   Create a Scratch program so you can enter the duration of a call, and the program can calculate and display the total cost of the phone call. Assume that the call will always be more than 3 minutes long, and that it will be entered as an integer.

String manipulation

4. Look at the program that follows, then complete the tasks listed.



- Work out what the value of FinalWord will be at the end of the program.
- Write a brief explanation of what the program does.
- Create a flowchart for this algorithm, title it correctly and add it to your algorithms file.

5. Create a program that asks for a user's name and then spells out the name by adding in dashes between all the letters (so if the user entered 'Suzy', the program would display 'S-u-z-y').

6. Create a program that will allow the user to enter a sentence and that will then display every third character of the sentence (e.g. 'The cat sat-on-the mat' will produce 'eas ea'. Note that spaces count as characters).

7. There is a word game where a player has to come up with a word starting with a letter that is the same as the last letter in the word the previous player used. The player who cannot do this, loses the game.

   Write a program to simulate this game. Improve the game by flashing the last letter of the previous word on the screen.

8. Every person who joins a specific company needs an account code to log on to the computer system. The account code is constructed as follows:

   - The surname is reversed – e.g. Naidoo becomes oodiaN.
   - The two-digit month number for the month the person was born in is then added to the end of this string (e.g. January is 01 and December is 12).

   Write a program that accepts a person's name, surname and date of birth (yyyy/mm/dd). It should then generate and display the account code.

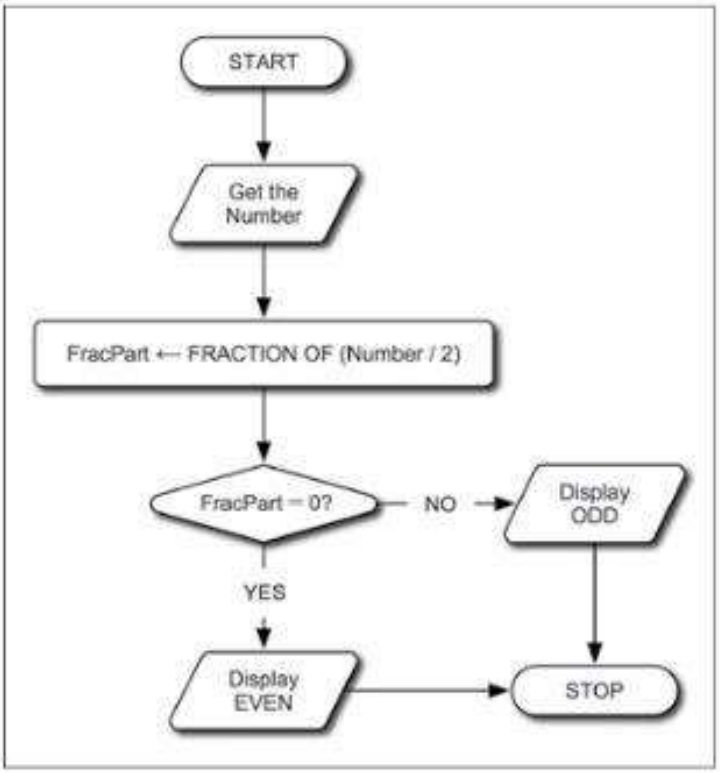
MODULE 1.4

Decision making

Introduction

You have learnt most of the programming structures that you need to be able to create entertaining and interesting programs. We have covered input, output, basic animation, repetition, variables, string handling and mathematical calculations. What we are going to do in this module is to use the programming structure which allows a computer to appear to be intelligent – the decision making structure.

You have already used decision making in some of the algorithms in Module 1.1.



Decision-making in Scratch

Scratch has a decision-making structure called an **If** statement. This statement can test for a condition and then execute sections of code based on whether that condition is true or false.

Creating a condition

A condition (also called a relational expression) is an expression which compares two values using a relational operator such as > and <.

The three relational operators available in Scratch are = (equal to), < (less than) and > (greater than). You can use these operators on numbers as well as strings in a condition. **Variables** or **calculations** can be used in a condition.

In Scratch a comparison block is used to create a condition. The blocks can be found in the **Operators** instruction group.

Examples of conditions	Result	
	True	ROTFL Cat says: This table contains examples of conditions and their results.
	False	
	False	
	False	
	True Scratch is not case sensitive when working with strings (it ignores the difference between uppercase and lowercase letters).	
	False Scratch compares strings by looking at one letter at a time. The first letter of the two words will be compared. Since these letters ('A') are the same, the second letter of each word – the 'd' and the 'l' - will be compared. The ASCII value of the letter 'd' is smaller than the ASCII value of the letter 'l'. The conclusion is that 'Adam' is smaller than 'Alf', even if 'Adam' contains more letters than 'Alf'.	
	False The ASCII value of the letter 'A' is smaller than the ASCII value of the letter 'E'.	
	True The ASCII value of the letter 'd' is greater than the ASCII value of the letter 'b'.	

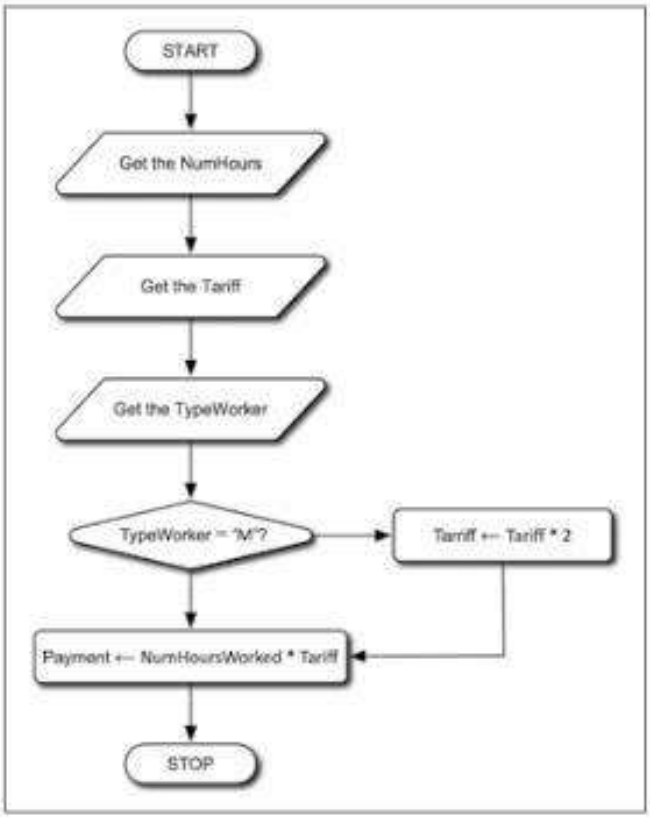
Two If statement program blocks are available in Scratch. Each one contains a single diamond-shaped opening which has to be filled with a condition.

The **[if]** block allows you to add instructions to the inner part of the block. These instructions will only be executed if the condition evaluates to true.

The **[if..else]** block allows you to add instructions inside the else part as well. The else part instructions will only be executed if the condition evaluates to false.

The [if] program block

The following are examples of **[if]** program blocks with conditions:



In Module 1.1 you had to create a flowchart for the solution to the following problem:

> A manager gets paid double the tariff per hour a waiter does. Enter the number of hours worked, a waiter's tariff per hour and the code indicating the type of worker. If the code is M (Manager), the tariff paid per hour must be doubled. Calculate and display the payment the worker should receive.

The Scratch solution to display the payment is as follows:

Note:

- In the second **[set]** block in the script, Tariff is set to the value supplied by the user.
- However, the tariff must be increased if a specific condition is met – the worker is a manager. The tariff will be increased – block multiplying Tariff by 2 – only if the condition in the **[if]** block is true. If the condition is false, the program block included in the **[if]** block will be skipped, and the payment will be calculated using the original value of Tariff.

1 Activity • Use [if] program blocks

1. A telephone company charges 50 cents per minute for a call. If the duration of the call exceeds 30 minutes, 45 cents is charged per minute. Enter the duration of a call in minutes. Calculate and display the cost as follows:
   Cost of call is Rxx and xx cents.
   Hint: Use the algorithm for integer division explained in Module 1.3.

2. Open the **Spot the Dot** program. When the program is executed, the user can drag the sprite (golden ball) to be positioned on one of the dots larger than the sprite. The sprite then announces that he is no longer positioned on the white background. Improve the program by adding [if] program blocks so the sprite will announce which colour he is positioned on.

3. Write a program to allow the user to enter any four digit number. Determine whether the number is a 'palindrome' or not. (A palindrome number is the same if read from left to right or right to left, e.g. 8558 or 9999.) Display a suitable message.

4. Create a program which will use a loop to generate 50 random integer values in the range of 1 and 700 (both inclusive).
   The program should determine:
   How many numbers generated are multiples of 8
   How many numbers generated are odd

Hint: To make it easier to test your program, first generate only 5 random values within the range of between 1 and 20. Once your program works with these values, you can change the values to those indicated in the given problem and your program should work for those values as well.

5. Open the **DroppingBlock** program. When the program is executed, a block falls from the top of the stage to the bottom.

   Improve the program by letting the colour of the block change after falling a certain distance. *Hint*: Use an **[if]** program block and the *y position* variable of the block sprite. Let the colour of the block change at least twice before it reaches the bottom of the stage.

6. Create a program which will read a base and power and calculate and display the value $base^{power}$.

7. A newspaper calculates the cost of an advertisement by charging 5c per character (spaces are not considered to be characters). Write a program where a user can enter an advertisement (a few short sentences), and the program has to calculate and display the cost for the advertisement.

8. You need to design a program for a courier company to calculate the cost of delivering a parcel. The cost is determined as follows:
   - The basic cost is calculated as R 1.23 per kilogram + transport cost per kilometre.
     - The transport cost per kilometre depends on the mode of transport: Road: R 0.15; Train: R 0.12; Air: R 0.36; Ship: R 0.25
   - If the customer wishes to insure the parcel, insurance cost is calculated as 11% of the basic cost.
   - If the customer wishes to have the parcel delivered as a high priority parcel, priority cost is calculated as 15% of the basic cost.
   - VAT must be calculated and added to the total cost (basic cost + insurance + priority). Round off the total cost to the nearest Rand.

   The sprite only needs to report on the total cost. All other values should be displayed on the stage using monitors.

   Use the following data to test your program:

Weight	Kilometers	Mode of transport	Insurance?	High priority?	Total cost
10	100	Road	Yes	No	R35
40.5	250	Train	Yes	Yes	R115
23.8	320	Air	No	Yes	R189
100	600	Ship	No	No	R311

9. Open the program **CountVowels**. Complete the script to count and then display how many times each vowel appears in the specified sentence.

The [if..else] program block

The program below demonstrates how an **[if ..else]** block and basic mathematical calculations, operators and functions can be used to solve a problem.

ROTFL Cat will ask you to enter a number. He will then tell you

- whether the number is even or odd
- whether the number is a perfect square or not
- what all the factors of the number are.

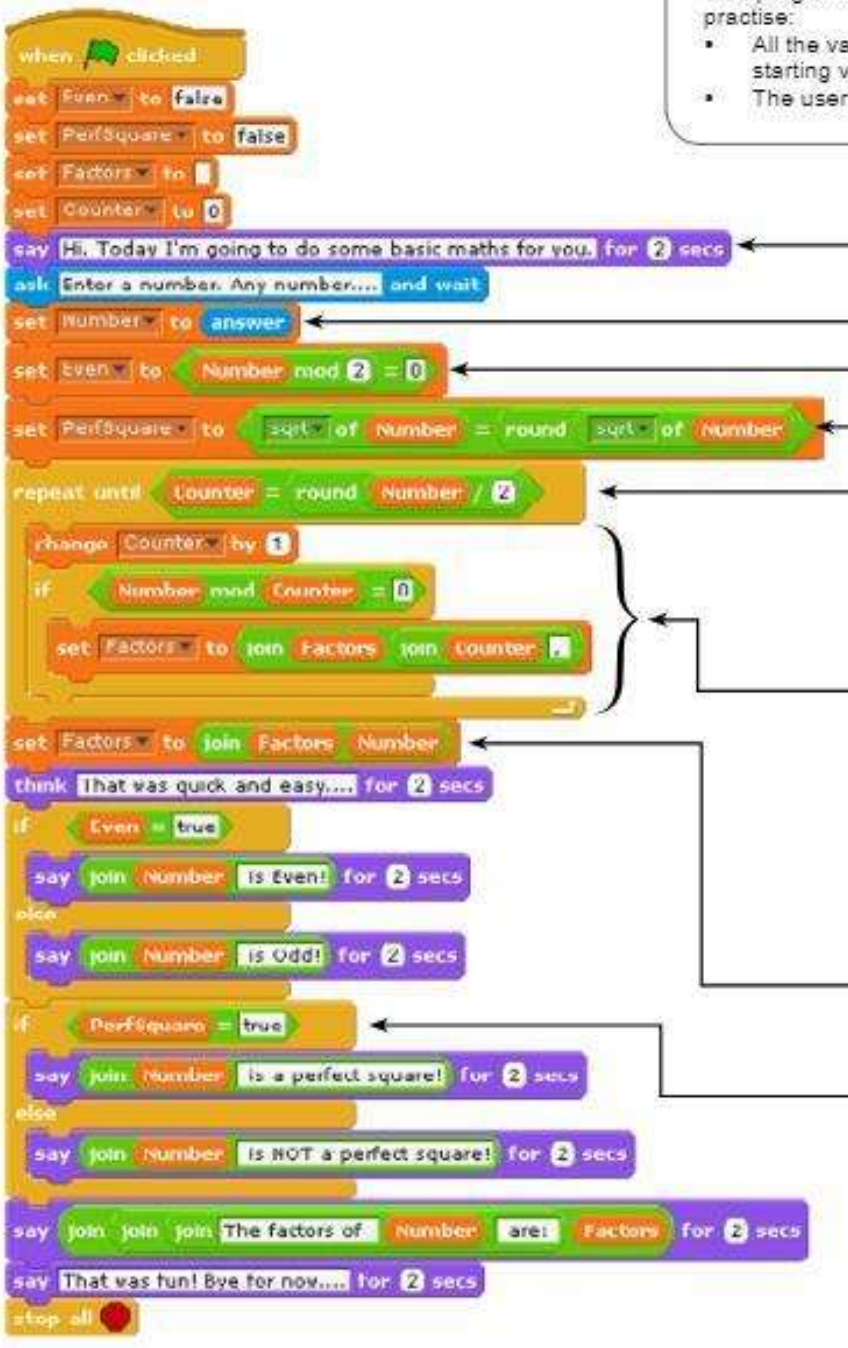
The following principles are used to make these decisions:

- An even number is a number that can be divided by 2 with no remainder.
- The square root of a perfect square has no decimal part (so it is the same as the rounded square root).
- A factor of a number divides into the number with no remainder.

Take a look at the program which follows:

This program illustrates good programming practise:

- All the variables are initialised (assigned a starting value).
- The user is informed what will happen.



Ask for input.

Store the input.

Calculate and store the value of Even.

Calculate and store the value of PerfSquare.

Create a loop to generate the factors. Any value bigger than half the number cannot be a factor of the number so the loop only goes up to half the number.

Test whether the counter divides into the number with no remainder.

If so, then add the counter value to the list of numbers that are factors (along with a comma and a space for neat formatting).

After the loop ends, add the number itself to the list of factors.

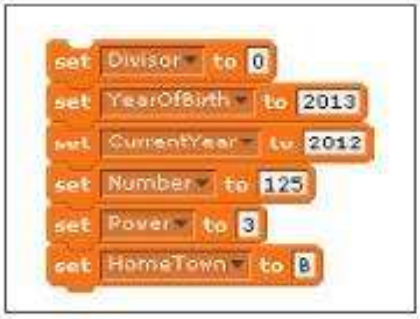
Finally display your results based on the contents of the variables.

ROTFL Cat says:

You will see extensive use of **[join]** blocks to make sure that the output from the program is neatly formatted and understandable to the user. It is the attention to detail such as the construction of informative input and output messages that makes the difference between a program that simply works and a program that is user friendly.

2 Activity • Evaluating Conditions

1. Assume the values displayed here are stored in the variables. Evaluate the following conditions, and write down what would be displayed. (You must do the calculations yourself if the answer involves calculations.)



a)

b)

c)

d)

e)

2. Now write down all the variables and indicate what possible values the variables should have so the message in the else part of each If statement will be displayed.

3. Finally, write out the messages that will be displayed using the variable values suggested in Number 2.

3 Activity • Use [if..else] program blocks

1. You run an Internet café. You need a program to calculate the money each client needs to pay.

   The cost is R6.00 per hour. If the client used the Internet for more than 3 hours, the cost is R4.00 per hour for the entire period of time the user spent on the Internet during this session.

   The client will enter the number of hours the Internet was used as an integer value. Calculate the amount of money the client must pay.

2. Look at the program doing basic mathematics that we discussed in the paragraph on the [if.. else] program block. A section of the program generates the factors of a number.

   How can this code be used to tell whether a number is a prime number or not? (A prime number is an integer that can only be divided by itself and by 1 without a remainder.)

   - Create an **algorithm** for determining whether a number is a prime number.
   - Write a **program** that reports whether a number is a prime number or not.

3. Adjust the program you wrote in Number 2 to report the largest factor of the number if the number is not a prime number.

4. Write a program to generate a random number between 1 and 10. Let the user guess the secret number until the correct number is guessed. Count the number of guesses it took to guess the number correctly.

5. Add program blocks to the program in Number 4 so the sprite will tell the user whether the number guessed is lower or higher than the secret number. For example, if the secret number is 8, and the user guesses 6, the sprite must display a message: 'Your guess is too low!'. If the user guesses 10, the sprite must display 'Your guess is too high!'.

6. Write your own Treasure Hunt program as follows:

   Place a treasure on the stage as well as a sprite looking for the treasure. When the sprite touches the treasure, it should become visible and the sprite has to call out that he has found the treasure.

   Expand the program by adding more treasures to find. Test whether all the treasures were found before the sprite can claim that he is the winner.

Using Boolean values

Boolean variables can be set using an **[if..else]** block, and they can be used in a condition.

Set the value of a Boolean variable

Let us look at a program which reads a password, and has to determine whether the password is correct. A variable, PasswordCorrect – a Boolean value - will be used. It must contain the value True if the password is correct and False if not. There are two ways in which the value of the variable PasswordCorrect can be set:

Method 1	Method 2
 An **[If..else]** block is used.	 Firstly, the condition is evaluated and then the result is assigned to the variable. If the result is true, the value True is stored in the variable PasswordCorrect. If not, the value False is stored in the variable. This is equivalent to the **[if..else]** block showed in Method 1.

Use a Boolean variable in a condition

The value of a Boolean variable can be compared to True or False in a condition. True and False are Boolean values and can be written in upper or lower case.

Compound Conditions

Logical operators

The combination of conditions allow you to create more complex decisions. This increases the so-called intelligence of your program. Let's first take a look at the logical operators which can be used with conditions.

ROTFL Cat says:

A **logical operator** operates on one or more conditions and gives a Boolean value as result.

Logical operator	Example of a condition
NOT operator NOT is the simplest of the logical operators. It takes the result of any other condition or Boolean value and inverts it – TRUE becomes FALSE and vice versa.	NOT (5 > 4) evaluates as FALSE 5 > 4 is TRUE NOT(TRUE) is FALSE (The inverse of TRUE is FALSE)
AND operator AND combines conditions so that only if ALL the conditions have TRUE as a result will the combined condition be TRUE. Condition with a TRUE result are printed in bold.	**(8 > 5)** AND **(8 < 10)**= TRUE because both conditions are TRUE **(8 > 5)** AND (8 < 7) = FALSE because one of the conditions is FALSE. (TRUE) AND (FALSE) = FALSE Name ← 'Jane' **(Name > 'A')** AND **(Name < 'M')**= TRUE want beide voorwaardes is TRUE. Name starts with the letter 'J'. The ASCII value of 'J' greater than the ASCII value of 'A', and smaller than the ASCII value of 'M. (Name > 'M') AND **(Name < 'Z')**= FALSE The ASCII value of the letter 'J' is smaller than the ASCII value of 'M'. Therefore (Name > 'M') =FALSE, and the combined condition is FALSE.
OR operator OR combines conditions so that if ANY ONE of the conditions evaluate to TRUE the combined condition will be TRUE.	**(8 > 5)** OR **(8 < 10)**= TRUE because both conditions are TRUE. **(8 > 5)** OR (8 < 7) = TRUE because one of the conditions is TRUE. Name ← 'Jane' ... **(Name > 'A')** OR **(Name < 'M')**= TRUE (Name > 'M') OR (Name < 'H') = FALSE because none of the conditions evaluate to TRUE.

Create relational operators not available in Scratch

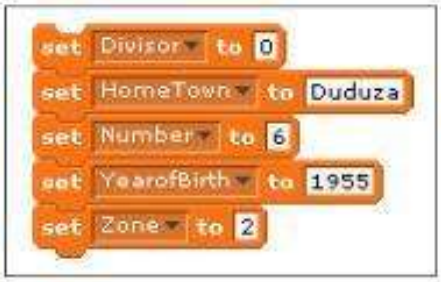
Some programming languages have additional relational operators: ≥ (greater or equal to); ≤ (less than or equal to); ≠ (not equal to). If you need to create conditions using these operators in Scratch, you need to make use of logical operators.

Here are examples of conditions and logical operators used to create the additional relational operators:

Mathematical condition	Scratch program and condition blocks
ExamPerc ≥ 80	
Score ≤ 29	
Grade ≠ 12	

4 Activity • Evaluating Combined Conditions

1. Assume variables have the values displayed here. Evaluate all the conditions and write down what will be displayed in a Scratch program. Also comment on the validity of the conditions.



a)

b)

c)

d)

e)

2. Now write down all the variables and indicate possible values the variables could have so the message in the else part of the If statement will be displayed

5 Activity • Use Combined Conditions

1. Students are selling tickets for entrance to a festival. They receive 10% of their ticket sales as payment. The price of student tickets is R50.00 per ticket and lecturers' tickets are R60.00 per ticket. If a student sells more than 10 lecturer tickets and more than 20 student tickets, he/she receives 12% of his/her ticket sales as payment.

   Write a program to enter the number of student tickets and the number of lecturer tickets sold. The program must calculate and display the total ticket sales and the payment the student will receive.

2. Consider the following algorithm. What would be stored in the variables **Result** and Y once the code has been 'executed'? (You do not need to write this program in Scratch.)

```
X, Y = Integers
C, D, Result = Boolean
X = 3
Y = 19
If ((YMOD5) > X)
     D = True
     Y = Y + 7
     C = False
Else
     D = False
     Y = Y + X
     C = True
Endif
C = not D or True
Result = D and C
```

3. Calculate the percentage a learner gets for a test if the learner attained a certain mark out of 60 for the test. Determine whether the learner has a distinction (≥ 80) for the test. Display the mark, percentage and suitable message.

4. The organisers of a singing competition use a grading system to see whether a contestant is through to the final round:
   - Judges as well as the public can cast votes.
   - A score out of 100 must be calculated for each contestant using the percentage of public votes, the number of judges' votes and bonus points. The score is calculated as follows:
     - The percentage votes from the public is converted to a number out of 60.
     - The number of votes from the judges (out of the 4) is converted to a number out of 30.
     - They score 10 additional bonus points if they receive more than 75% of the public votes or more than 75% of the judges' votes.
   - To qualify to be in the final round, a contestant must have a score of at least 75 and have at least two judges' votes.
   - Input to the program:

     Number of persons from the public who voted; Number of persons from the public who voted for this contestant; Number of judges who voted for this contestant.
   - Create a flowchart for this program. The program must ask all the necessary values (input) from the user, then calculate and display the final score.

   Now create the program in Scratch.

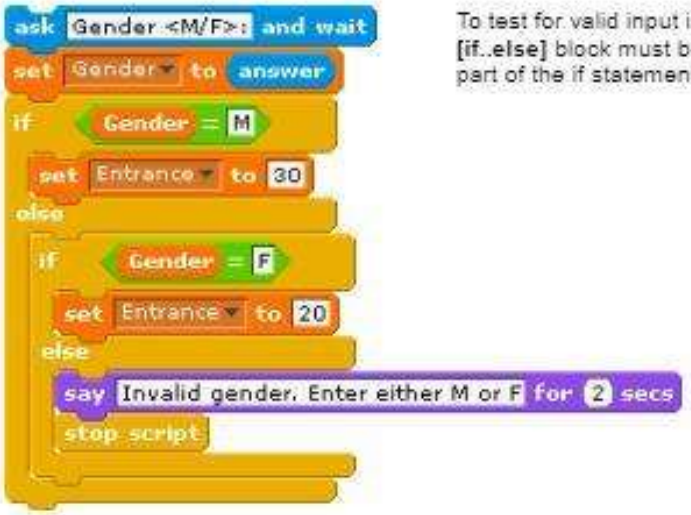
Nested [if] blocks

A decision structure can be used inside any part of another decision structure. This is called **nesting**.

Let us look at the following scenario.

In a program a user has to indicate his/her gender by entering either the letter 'F' or the letter 'M'. Based on the letter entered, the entrance fee is set to a specific value. However, the user may enter an invalid letter such as 'D'. In this case the following code will not be effective:

Any letter other than 'M' will have as result that the person is assumed to be female (and will pay R20 entrance fee). It is a good programming principle to test for valid input and inform the user whether the input was invalid.



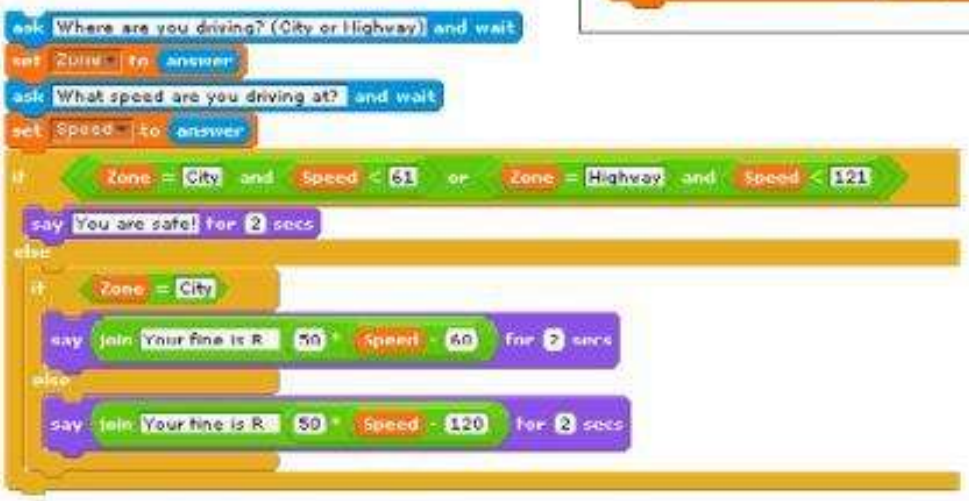
To test for valid input in this case, another **[if..else]** block must be used in the else part of the if statement.

Later in this module we will show you how to use a program block so the program will repeatedly ask for a letter to be entered until it is either 'M' or 'F'.

6 Activity • Evaluating Combined Conditions and nested [if] blocks

1. Assume variables contain the values displayed here. Evaluate all the conditions which follow and write down what would be displayed in a Scratch program (you have to do the calculations if the answer requires calculations to be done).

a)



b)

7 Activity • Use nested [if] blocks

1. Create a program where a sprite will ask the user how many SMS messages they sent in a month and the cost per SMS message. The user will then be required to indicate whether they are on a cell phone contract or not. The first 20 messages will be free of charge for a contract. The program must calculate the total cost of the SMS messages sent.

2. Write a program where the user can enter a year, e.g. 2003, and the sprite has to determine whether the year is a leap year or not. A leap year is determined as follows: If the year ends in 00 (e.g. 1900 or 2000), it is a leap year if it can be divided by 400 without a remainder. If it does not end in 00, it is a leap year if it can be divided by 4 without a remainder.

3. Write a program where the user enters a date in the format yyyy/mm/dd. The sprite has to determine whether the month is valid (a number ≤ 12), and whether the day is valid. A day is valid if the number of days is ≤ 31 for month numbers 1; 3; 5; 7; 8; 10 and 12. For month numbers 4, 6, 9 and 11 the number of days should be ≤ 30. For month number 2 (February), you need to determine whether the year is a leap year or not. For leap years the number of days for month number 2 should be ≤ 29, and for non-leap years ≤ 28..

   This program may contain many nested [if..else] blocks. Remember to make use of Boolean values and the 'quick' way to set the value of a Boolean variable.

8 Activity • Use nested [if] blocks and Compound Conditions

1. Enter a name and date of birth in the format dd/mm/yy (the user needs to enter a '0' as the first digit if the day or month number is a single-digit number e.g. 06/01/89).
   Create a string to display the date in the format dd/mm/yyyy. E.g. 06/01/1989.

   If a year between 0 and 29 (both included) is used, the date of birth was in the 21st century, so you should add the digits '20' to the front of the year value. If the year value is between 30 and 99 (both included), the date of birth is in the 20th century, which means '19' should be added to the front of the year value. E.g. 20/06/45 must become 20/06/1945.The date 23/12/01 must become 23/12/2001.

2. Expand the program you wrote in Number 1. The program should also display a message if the person was born on a leap day (29 February).

3. Expand the program you wrote in Number 2. The program should also determine and display the season in which the person was born.
   - December until February is summer.
   - March until May is autumn.
   - June until August is winter.
   - September until November is spring.

ROTFL Cat says:

You should realize by now that

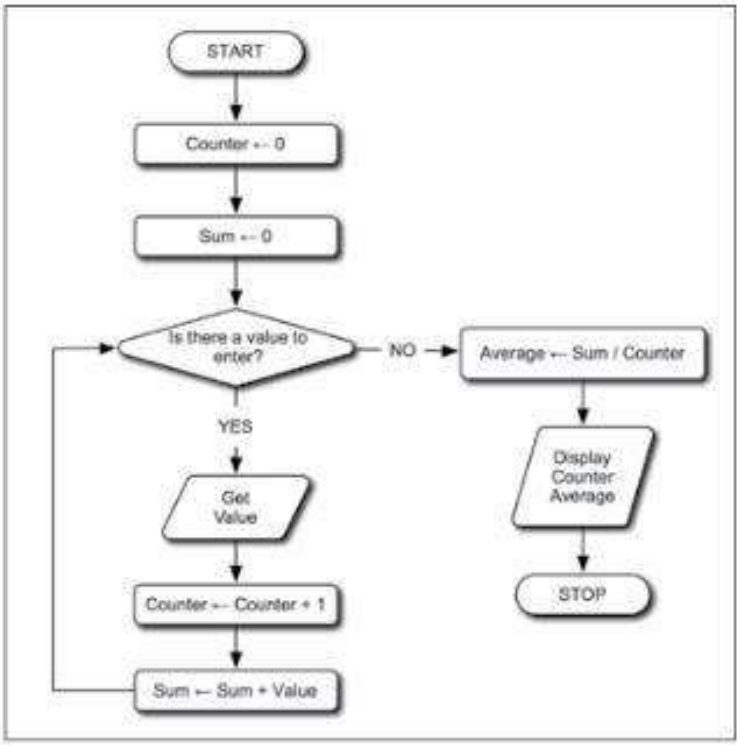
- the use of combined conditions and nested instructions allows your program to gain a great deal of 'intelligence'.
- the more 'intelligence' you build into a program, the more complicated the program becomes to write.
- the more 'intelligence' a program has, the more useful and flexible it becomes.
- you can use decisions to make sure that the user inputs meaningful data so that your program works properly without the user being able to mess it up with bad data.

Decide when to terminate the loop

In previous modules, we have done many examples where we have used the **[repeat ( )]** program block to repeat instructions a certain number of times. Another program block often used for the repetition of instructions is the **[repeat until]** program block in the **Control** instruction group. It uses a condition to determine whether the instructions in the loop should be executed, or whether the loop should terminate. You have already used this block to control movement. For example:

We are now going to look at examples of programs where variables and values are used in the condition of the **[repeat until]** program block.

In Module 1.1 we discussed a flowchart where a number of values could be entered by the user, and when the user indicates that there are no more values to enter, the average of the values entered is calculated and displayed. The Scratch program for this flowchart can look as follows:



In the **[repeat until]** loop, the loop will terminate when the condition is **true**. The variable called **Decision** controls the **[repeat until]** loop.

If the variable Decision contains the value 'n', the condition is true and the loop will be terminated.

Any other value for Decision will let the condition evaluate to false, and therefore let the loop continue.

Scratch will evaluate the condition in the **[repeat until]** block.

- If the condition is false, the instructions inside the **[repeat until]** block (loop) will be executed
- If the condition is true, the loop will terminate. The remainder of the script following the **[repeat until]** block will then be executed.
- There has to be a possibility of the condition controlling the loop to become true. Therefore one of the instructions inside the loop must allow for the content of the variable used in the condition to be changed.

9 Activity • Analyse a program

This activity relates to the previous program that calculated the average of a series of values:

Open the program **CalcAverage** and experiment by adding or deleting program blocks while you are trying to determine the answers to the questions that follow:

1. What is the minimum number of values a person will ever be able to enter?
2. How will a user indicate that there are no more values to enter?
3. Why is it necessary to set the value of average to 0 at the beginning of the program?
4. Why is it necessary to repeat the **[ask]** and **[set]** blocks before the end of the **[repeat until]** block?
5. What will happen if the user types the value 'n' to the first question?
   Add additional code to prevent this error.

ROTFL Cat says:

Pay attention! Potential problem here!!!!!!

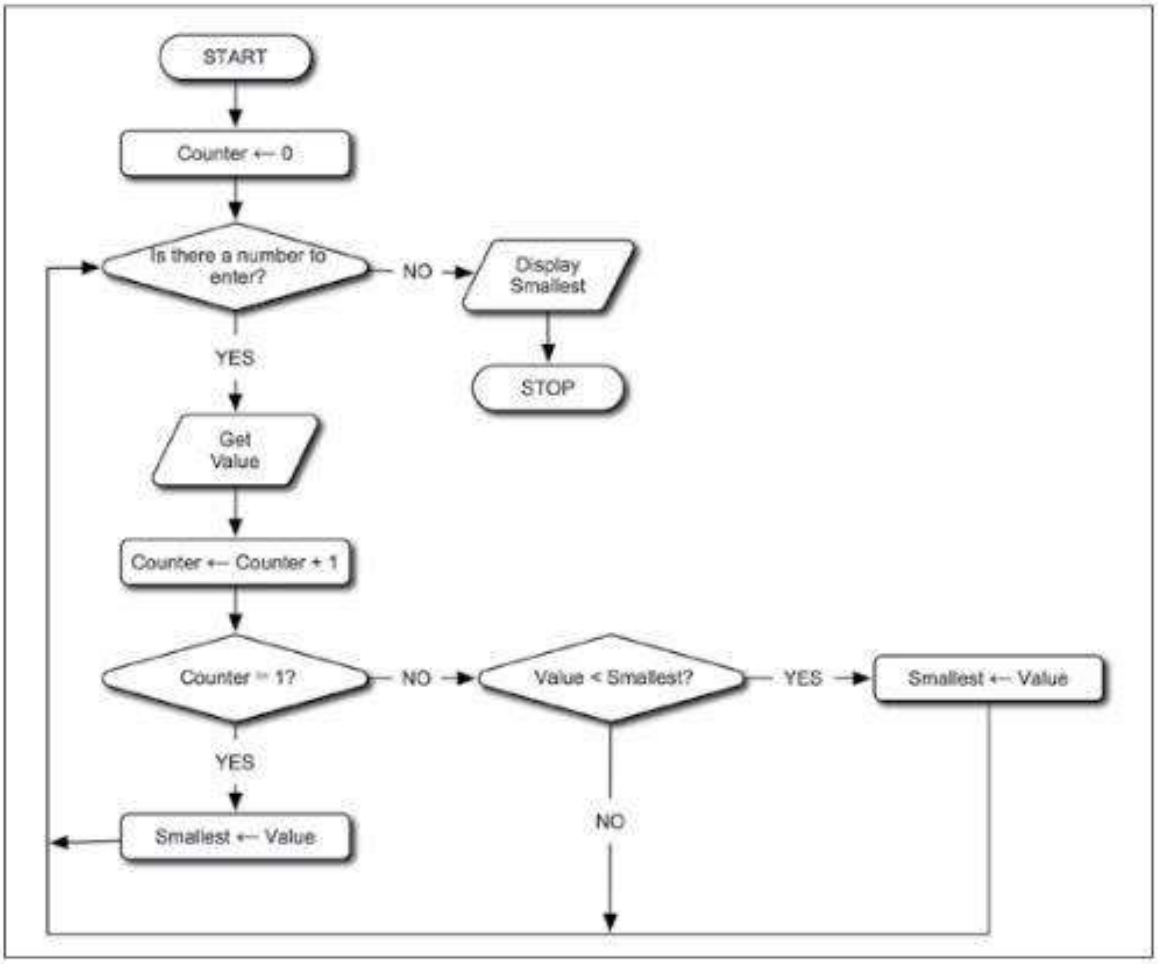
If you don't plan how your repetition will end, your program will 'hang' (i.e. your program – maybe even your computer – will appear to stop working).

What has actually happened is that your program has gone into an ***infinite loop***. Computers don't think or get tired, so it keeps on doing the repetition until the computer breaks down, the power is interrupted or the world ends (whichever comes first). You don't want this to happen to your program!

PS: Computer people have a great sense of humour. What company has its headquarters at 1 Infinite Loop, Cupertino, California?

10 Activity • Write programs using repetition

1. Open the program **CatWalk**. Why does the program never stop executing? (You can see this from the white frame around the script.) Correct this error.
2. Open the program **Throw dice until 6**. The program is supposed to keep on 'throwing' a dice. (It uses the **[pick random]** block to simulate the throw of a dice).
   Add program and operator blocks to let the program count and report on the number of times the dice was thrown before a 6 was thrown.
3. Turn back to the paragraph on the nested if statement. Look at the example that was used to explain the concept. Add a **[repeat until]** block and program blocks to the script to repeatedly ask a user to enter a character until either the character 'm' or 'f' is entered.
4. An advertising company charges 20c per word to place an advertisement. Write a program to let a user enter a sentence, then count how many words there are in the sentence. Assume that each word is followed by one space. *(Tip:* Add a space to the end of the sentence so the last word can be detected). Let a sprite display the amount to be paid.
5. The flowchart that follows determines the smallest value of a series of values that a user enters. Write a Scratch program for this algorithm.



Decide when to enter the loop

Lastly we want to discuss the **[forever if]** program block in the **Control** instruction group. This loop also uses a condition to determine whether the instructions in the loop should be executed.

11 Activity • Explore the [forever if] block

Open the program **ForeverFan**. Execute the program, and see what the effect is if the mouse button is held down.

1. What will the value of the condition in the if part of the loop be when you click and hold the mouse button?
2. What will the effect be if you replace the [forever if] block with a [wait until <mouse down>] block?
3. Will you be able to add a program block such as [say] at the end of the [forever if] block?
4. How do you stop this program?

Scratch will evaluate the condition in the **[forever if]** block

- If the condition is true, the instructions inside the **[forever if]** block (loop) will be executed.
- If the condition is **false**, the block will not terminate, it will continuously re-evaluate the condition. If the condition becomes true again, the loop will be entered (instructions inside the loop will execute).

The only way to stop the program is by clicking on the button **Stop everything** (the red dot).

12 Activity • Use [forever if] to create animation

Open the program **Paddle and Ball**. It has two sprites.

The script for the Paddle sprite only makes the paddle move horizontally on the screen based on the movement of the mouse pointer.

The script for the Ball sprite makes the ball move up and down the screen.

Write another script for the Ball sprite to make the ball turn a random number of degrees every time it touches the paddle (degrees must be between 160 and 210 both included). When the two sprites touch one another, the ball must display the next costume. (A number of costumes are available in the program. The effect will simulate a spinning ball.)

ROTFL Cat says:

Did you notice? The **[repeat until]** and the **[forever if]** statements work in opposite ways:

- **[repeat until]** executes the instructions inside the loop when the condition is **false.**
- **[forever if]** executes the instructions inside the loop when the condition is **true.**

13 Activity • Save the mouse

Open the program **Cat_n_Mouse**. In this program the Cat sprite follows the Mouse sprite. As soon as the Cat sprite comes too close to the mouse, the mouse moves to another position. However, sometimes the cat 'catches' the mouse by touching it.

- Change the program to make use of a counter. Increase the counter each time the cat touches the mouse.
- Change the program using any program instructions you know, so the mouse will be able to 'run away' faster, or in a direction that will prevent the cat from touching it. Have a competition in class to see whose program will have the least 'catches' in a certain period of time.

A 'stopwatch' called the timer can be used to display the number of seconds that elapsed in a program. Do the following to activate the timer:

Add the following script to the stage. Tick the CheckBox next to the variable **timer** in the **Sensing** instruction group:

MODULE 1.5

Interaction between sprites

You would have noticed that a sprite is only capable of 'controlling' itself, and working with its own data. Program blocks such as **[show]** and **[hide]** can only make the sprite itself appear and disappear. When a sprite does a calculation, he uses only the variables available in his own **Variables** instruction group. In this module we are going to learn how to let sprites communicate with one another. There are two types of communication between sprites:

- A sprite can broadcast a message to other sprites to let them know it is their 'turn' to execute a script.
- Different sprites can work with the same variables – they can share data.

Broadcasting and receiving messages

In Scratch, scripts can be activated when an ***event*** takes place (for example, when the flag is clicked or the spacebar is pressed). The user is responsible for these events.

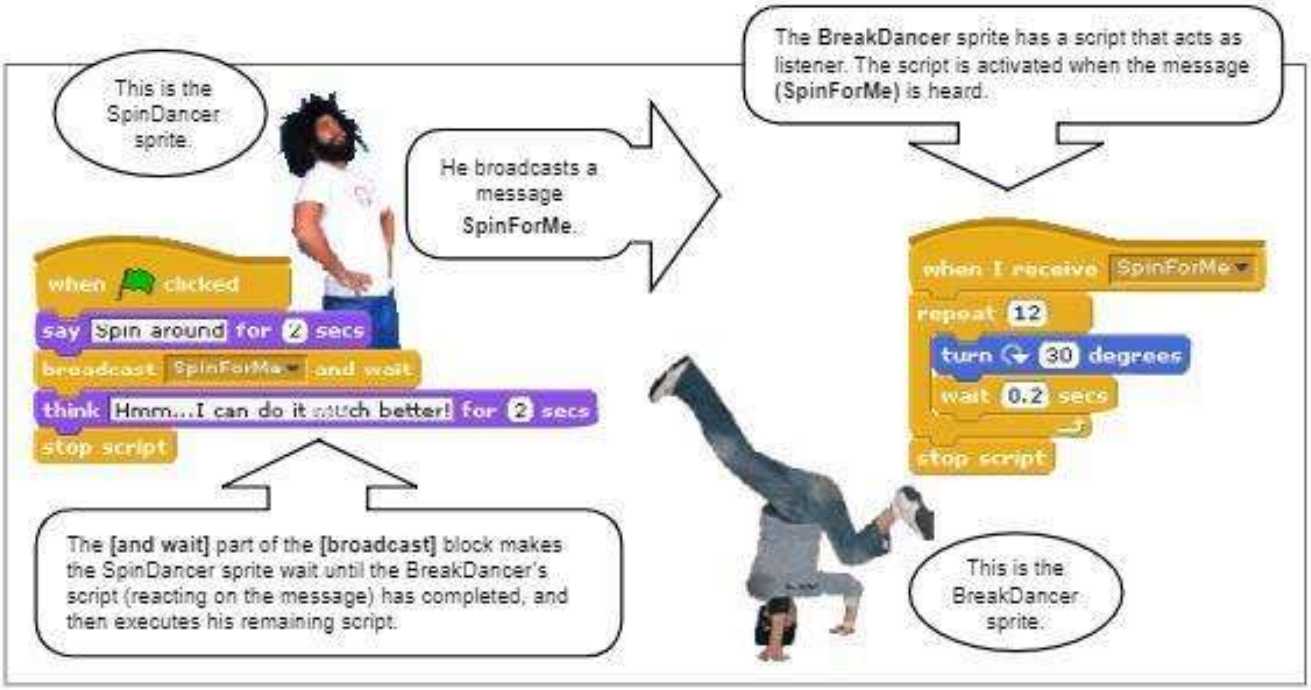
An object (a sprite or even the stage) can also activate scripts while the program is running. This can be done as follows:

The object **broadcasts** a specific **message** using a **[broadcast]** program block.

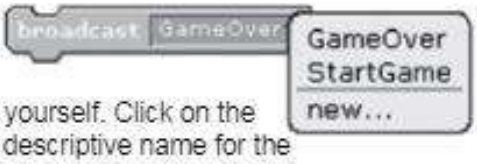
→

One or more objects can act as listeners for the specific message by means of the **[when I receive]** program block. When the message is broadcast, the script(s) of all the listener(s) who waited for the message will be executed.

This is an example of one sprite that broadcasts a message, and the script of another sprite that reacts on the message:



Notes:



- The blocks related to broadcasting and receiving messages are found in the **Control** instruction group.
- You can choose the name of the message you want to broadcast yourself. Click on the down-arrow in the **[broadcast]** block. Choose **new...** and enter a descriptive name for the message you want to broadcast.
- One or more scripts can react to the same message.
- A sprite can react on a message it sent itself.
- The **[broadcast]** block only broadcasts the message, and then continues with the script of the broadcaster.
- If the **[broadcast and wait]** block is used, all the listener scripts that react to the specific message will first complete their scripts in reaction to the message before the remaining script of the broadcaster sprite will be executed.

The following activities contain code which you will probably be able to use in projects where you will have enough time to experiment with animation and interaction between sprites.

1 Activity • Create a cartoon

Open the program called **Movie**. The following scripts have already been written.

Three backgrounds:
Stars
Desert
Bush

5 scripts	4 scripts	4 scripts
		
		
		

Write scripts for the Starfish and Lion sprites to complete the cartoon.The cartoon should also include dialogue for the Desert and Bush backgrounds.

Note:

Both the Lion and the Starfish sprites sent themselves messages (Shake and Spin) to run some of their own scripts that exist already. This has the advantage that, should they want to repeat the same sequence of steps in another script, they only need to broadcast a message, and do not have to repeat the code.

2 Activity • Create a dialogue

Write a *Scratch* program containing at least 2 sprites. The sprites have to take part in a dialogue. Each one should talk at least 3 times during their conversation. Use your imagination – have them tell a knock-knock joke, or be participants in a quiz or any other situation where characters talk to one another.

3 Activity • Throw away the paper

Open the program **PaperIn_or_Out**. The stage contains a rubbish bin. Each time you click on the flag, a piece of paper is thrown towards the bin. (The **[pick random]** block is used to determine the position where the paper should land.) There are two counters on the stage already (Missed and Hits).

a) Add code to the scripts of the stage to also display the percentage of efforts that landed in the dust bin.

b) Add more **[broadcast]** messages so the sprite which tests whether the paper has landed in the bin (HitArea), becomes smaller as soon as the number of hits is more than 60%. (It should therefore become more difficult to hit the bin.)

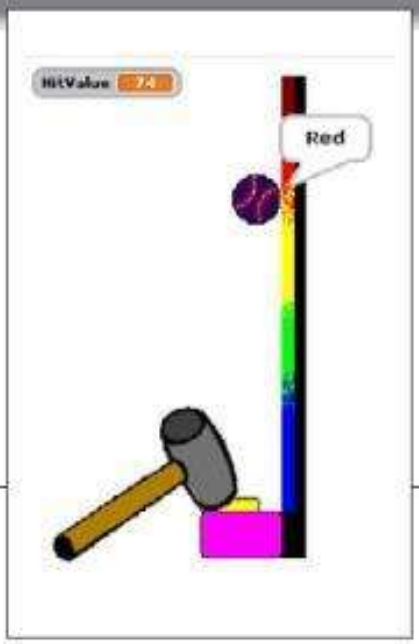
4 Activity • Create a traffic light

Open the **TrafficLight** program. Complete the *scripts* for the stage and all three sprites so the red, yellow and green lights will be switched on and off alternately.

Hints:

- The stage has to broadcast **[broadcast and wait]** messages to turn on the lights.
- As soon as the program starts, all the lights first have to be switched off. Use a message such as **[broadcast switch off and wait]** to accomplish this.

5 Activity • Measure your power



Open the **HammerBall** program. The game works as follows:

- A ball is positioned on a platform, which is attached to a colour coded rod. The rod is blue at the bottom, followed by green, yellow, red and indigo (at the top).
- When a hammer hits the platform, the ball will bounce upwards. The harder the hammer hits the platform, the higher the ball will bounce.
- The ball has to display the name of the colour on the rod it has reached and also play the sound with the same name. (The sounds have been loaded already.)
- Complete the *script* of the Ball sprite according to the following pseudo code:

***Script* of the Ball sprite**

Wait for the HIT message

Face the top of the stage

//Simulate how hard the hammer will hit.

Use the variable HitValue to save a random number between 1 and 100 (both inclusive)

// Calculate the height of the bounce of the ball. Do not use any other value than 2.5. The boundaries of the range of colours have been calculated based on this value.

PowerValue = HitValue x 2.5

Move upwards based on the value of PowerValue*// To let the ball bounce upwards*

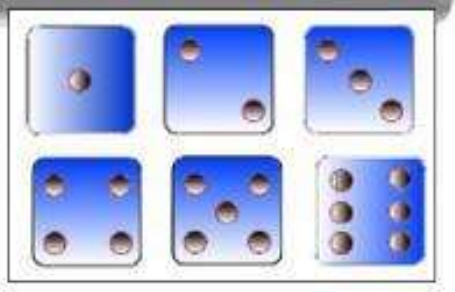
Determine the colour of the rod reached according to the height of the bounce of the ball

Say the name of the colour

Play the sound with the same name as that of the colour

The boundary values of the colours are as follows:
0..20: Blue (the name of the sound is 'blue')
21..40: Green (the name of the sound is 'green')
41..60: Yellow (the name of the sound is 'yellow')
61..80: Red (the name of the sound is 'red')
81..100: Purple (the name of the sound is 'indigo')

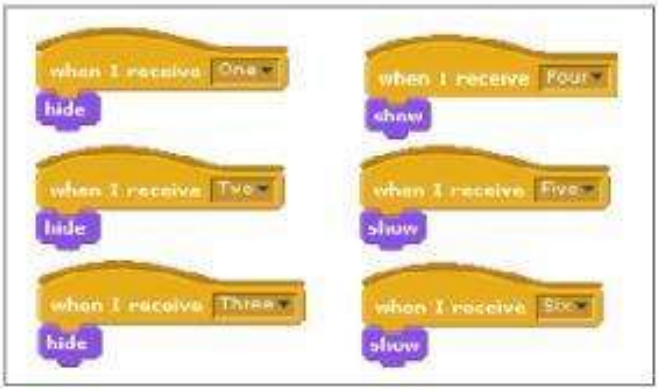
6 Activity • Display a die



Each surface of a die contains a certain number of dots in a pattern. Each pattern is made up of dots in specific positions. There are seven possible positions a dot can take. We can indicate the position of each dot by referring to the column (to the Left or to the Right) it can be positioned in, as well as the row (Top, Middle, Bottom). That means that the dot in the left hand column and in the bottom row will be referred to as LBottom.

Open the **Show_the_die** program. It contains a number of sprites, each representing a dot in a specific position on a die. The sprites have been named as explained before, e.g. RMiddle, LTop, etc. The dot in the middle has been named Centre.

The stage already contains a script generating a random number between 1 and 6 and broadcasting a message indicating what number has been 'thrown'. Create scripts for each of the dot sprites, so that the dots displayed on the die will match the number 'thrown'. This is an example of the scripts required for the dot on the left-hand side at the bottom (LBottom).



Two sprites Communicate by sharing data

You will remember that the following dialog box appears when you create a variable in Scratch:

Originally you were taught to select the **For this sprite only** option. When you do this, only the current sprite will have access to the variable (can change the value or show the variable).

Sometimes, however, it is necessary for more than one sprite to have access to the same variable. Let us look at the following program – **FastFoods** (you can open the program while you read the explanations):

The program should do the following:

1. Count how many meals have been ordered. The variable **CountItems** is used for this.
2. Calculate the total cost for all the meals. The variable **TotalCost** is used for this.

The **stage** does the following:

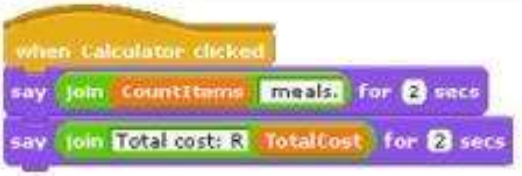
- When the space bar is pressed, both variables are set to zero.

The **CheesePuffs** sprite does the following:

- When you click on the sprite, it asks how many cheese puff meals should be bought.
- The cost of one meal is R5.00. The cost of all the cheese puff meals is calculated, and the value is stored in CostPuffs.
- The cost of the cheese puff meals (CostPuffs) is added to the total cost (TotalCost).
- The number of cheese puff meals that were bought is added to the total number of meals that were bought (CountItems).

Note: The variables **CountItems** and **TotalCost** that are changed by the CheesePuffs sprite, are the same variables that were set to zero by the stage.

The **Calculator** sprite does the following:



Note: The Calculator sprite must also have access to the two variables **CountItems** and **TotalCost**, because it must report their final values.

Both the Fruit sprite and the ToffeeApple sprite must do the same type of calculations as the CheesePuffs sprite. It is therefore necessary that all the sprites, as well as the stage, will have access to the two variables **CountItems** and **TotalCost**. To manage this, you must indicate that the variable should be accessed by all the sprites when it is declared (click on the 'For all Sprites' option).

When a variable is declared in this way, the following will happen:

- The variable becomes part of the stage (even if the stage never actually uses the variable). The variable name will now be reflected in the **Variables** instruction group of the stage. For the FastFoods program, this is what it looks like:

- The variables appear in the **Variables** instruction group of each of the other sprites on the stage. If a certain sprite also has variables specific to the particular sprite, a line will appear to separate the two types of variables. The variables applying to all the sprites will appear above the line, and those that can be used by the specific sprite only will appear below the line. This is how the variables of the CheesePuffs sprite are indicated:

When a variable belongs to the stage (can therefore be accessed by all the sprites on the stage), it is known to have **stage scope**.

When a variable only belongs to a certain sprite, it is known to have **sprite scope**.

It is easy to remember the difference between stage scope and sprite scope: When furniture is put on a stage, all the actors can see and use it. Similarly, variables belonging to a Scratch stage (declared on the stage) can be used by all the sprites.

When one actor has a piece of paper in his hand, only he can read from it and none of the other actors can do so. Similarly, when a variable is declared for one sprite only, it can only be used by that specific sprite.

7 Activity • Complete FastFoods

Open the program **FastFoods**. Declare all the variables and complete the scripts for the Fruit sprite and the ToffeeApple sprite.

8 Activity • Double 6

Open the program **Double6**. Declare the variables and complete the scripts for the program to do the following:

- The ROTFL Cat sprite must ask how many times the dice must be thrown.
- Both the dice must be thrown simultaneously for this number of times. (Use **[pick random]** to simulate the throwing of the dice). The dice must display the costume which will match the number that has been 'thrown'.
- ROTFL Cat must count how many times both the dice have 'thrown' a 6, and report this number at the end of the program.

Use variables with a *stage scope* only when it is absolutely necessary.

9 Activity • Selling Tins

Open the program **TinCan**. A school is using this program to keep track of their tin recycling project. For each tin that a learner contributes to the project he receives 20c and the school receives 15c.

The program already contains three sprites, - a 'machine' and two buttons - a green button and a blue button.

The program must do the following:

- Allow a learner to indicate that he/she has contributed one tin to the project. This is done when the user clicks on the green button.
- Count how many tins the learner has contributed (while the green button is clicked it indicates that it is still the same learner making the contributions). Use a monitor on the stage to display the number of tins that the learner has contributed.
- Calculate the total number of tins that the school has received. This total must also be displayed on the stage using a monitor.

Clicking on the blue button is an indication that the learner has entered all his tins. The program must then do the following:

- Calculate and display (**[say]**) the amount of money that the learner must receive.
- Calculate and display (**[say]**) the amount of money that the school will receive for this learner's contribution.

During the running of the program, the total income for the school must be displayed on the stage using a monitor.

Use variables with a *stage scope* only when it is absolutely necessary.

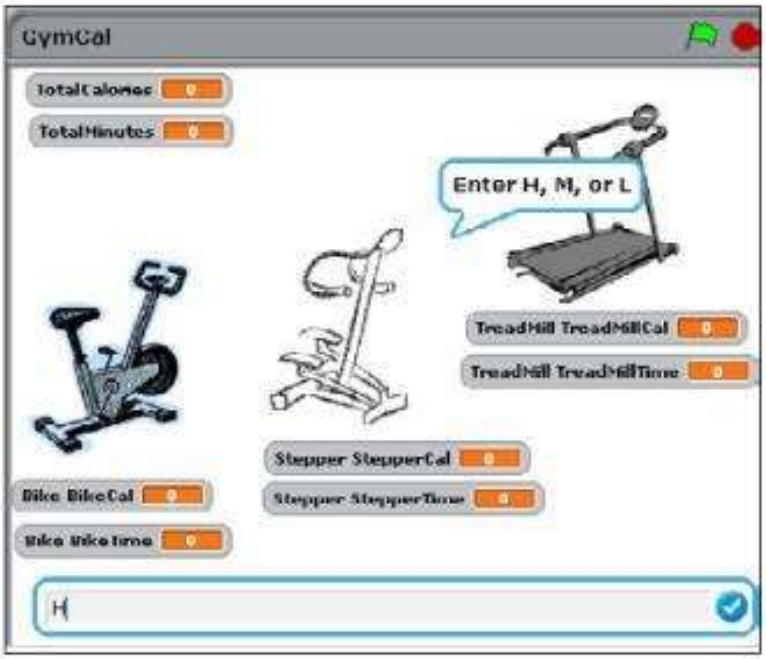
10 Activity • Exercise in the gymnasium

Open the program **GymCalc**. The program already contains three sprites representing three different exercise machines available in the gymnasium. Complete the program to do the following:

- The user will click on each machine he/she plans to use. For each machine the level of intensity at which the machine will be used should be entered, as well as the time he/she will spend on the machine.
  - The level of intensity can be low, medium or high.
  - The time should be entered in minutes.
- The user may use different machines on different levels of intensity for different periods of time during the same session.
- Each machine has the potential to burn a specific number of calories in one hour at a specific level of intensity. The intensity levels and the associated number of calories burnt per hour are reflected in the following table.
- Calculate the number of calories a person will burn during one exercise session, as well as the total amount of time he/she will spend in the gymnasium.

Bike		
Low intensity	Medium intensity	High intensity
800	1000	1600
Stepper		
Low intensity	Medium intensity	High intensity
1200	1700	2000
Treadmill		
Low intensity	Medium intensity	High intensity
800	1100	1500

Some of the possible output screens for this program are shown below.



When the green flag is clicked, the variables must be reset to zero.

MODULE 1.6

Lists

In a previous module we wrote a program to calculate the average of a series of numbers. Each time we entered another number, the value of the previous number was overwritten. Eventually the average could be displayed, but the numbers themselves were not available anymore. Let us say we want to write a program to calculate the average, and then count how many of the numbers are larger than the average. In other words, we want to use the original numbers without having to enter them again. In this case the data structure (the variable) we have been using is not adequate.

All programming languages have a data structure available that allows you to store values in a numbered list. In some programming languages this data structure is called an array. In Scratch it is called a List. If we use this data structure, all the values we enter will be stored in the List. We can then 'read' through the complete list at the end of the program to determine which values are greater than the average, because the values will still be available.

Representation of a list

You can represent a list containing a few numbers as follows:

The number of each item in the list is called the **index** of the item. The first item's index is 1, the second 2, etc.

	Numbers
1	67
2	89
3	76
4	45
5	74

The values in the list are called **elements** in most programming languages. Scratch refers to them as the **items** in the list.

Here's a quick summary of what you need to know about lists in Scratch:

- A list has one name (such as Numbers) but can contain multiple items in numbered boxes.
- Items can be added to or deleted from the list.
- You can replace an item in the list.
- Lists can contain values of different data types.
- You can use the value of a single item by referring to the index of the item in the list. The item can then be used in the same way as a normal variable.

Scratch differs from other programming languages again in this instance. The lists (arrays) of many of the other languages may only contain data of the same type, e.g. a list containing only integers, or a list containing only strings.

Create a new list

A list is created in the same way as a variable.

- Click on the **Variables** instruction group.
- Click on **(make a list)**.
- Indicate whether the list must be available to all sprites, or only to this sprite.

As soon as the list has been created, the program blocks used to manipulate items in the list become available.

Program blocks for lists

The following program blocks are available to work with items in lists:

add thing to WinnerNames	The **[add]** instruction adds an item to the end of the list. If the list contained 5 items then after this instruction the list will contain 6 items and the last one will be what you have just added.
delete 2 of WinnerNames	The **[delete]** instruction deletes the specified item from the list. The items that followed on the one removed will all move back so the list will have no empty items.
insert Tusi at 3 of WinnerNames	The **[insert]** instruction works like the **[add]** instruction, except that it puts the new item at a specified position. All the items that follow are moved on one position.
replace item 1 of WinnerNames with Simon	The **[replace]** instruction changes what is in the block at the specified position to what you want to put in that position.
set Winner to item 1 of WinnerNames	The **[item of]** instruction accesses the item at the position.
length of WinnerNames	The **[length]** instruction tells you how many items are in the list. You can use this to process the whole list, e.g. by using a loop.
if WinnerNames contains Simon say Simeon is one of the winners!	The **[contains]** instruction searches the whole list to see if it contains what you are searching for. It must be an exact match (it will not find part of a string – only matches whole strings).

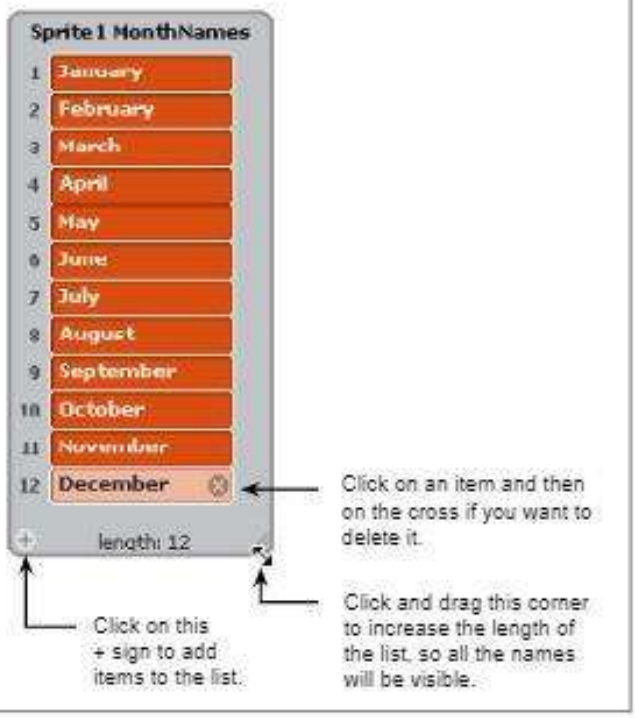
Store items in a list

There are many ways to store items in a list. We are going to use one program as example to demonstrate the different ways.

Suppose we want to write a program where the number of a month is received as input, and we want to display the name of the month. We are going to store the names in a list MonthNames, and then simply use the number of the month to 'look up' the name in the list.

Add items to a list before the program executes

As soon as the list has been created, an empty list will appear on the stage. Click on the '+' sign in the corner of the list – you will then be able to enter items in the list. Add the names January to December to the list.

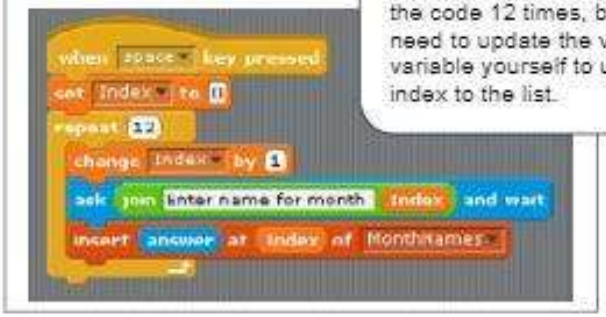


Code for a program to 'convert' the number of a month to the name of the month can look as follows:

Scratch retains the values of a variable after the program stopped executing. Even when you close the program and open it again later, the list containing the names of the month will still be available.

Values entered by the user

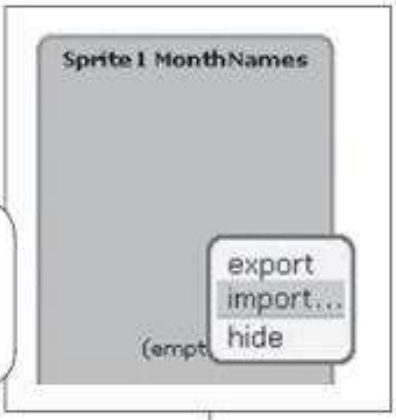
You can write code to enable the user to enter the values to be stored in a list. The values only need to be entered once.



The **[repeat]** block will execute the code 12 times, but you need to update the value of a variable yourself to use as an index to the list.

Read values from a file to the list

You can store the names of the months in a text file – one name per line – and import them from the text file to the list. Right-click in the grey area of the list, and choose **import...** An **Import List** dialogue box will appear. Choose the text file that contains the names of the months. Each line in the text file will become an item in the list.



You can also store the values in a list in a text file by choosing **export**.

To create a text file

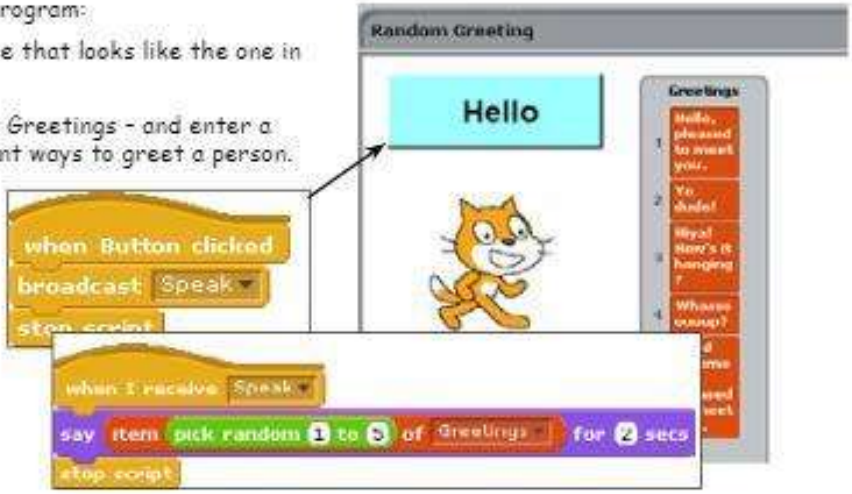
Open any text editor such as Notepad.

Type the names of the months and store the file in the same folder as the program. The extension of the file should be .txt. Close the file and the text editor.

1 Activity • Randomly choose a message from a list

Write a new Scratch program:

- Create an interface that looks like the one in the screen shot.
- Create a new list - Greetings - and enter a number of different ways to greet a person.
- Write the script for the Button sprite.
- Write the script for the Cat sprite.



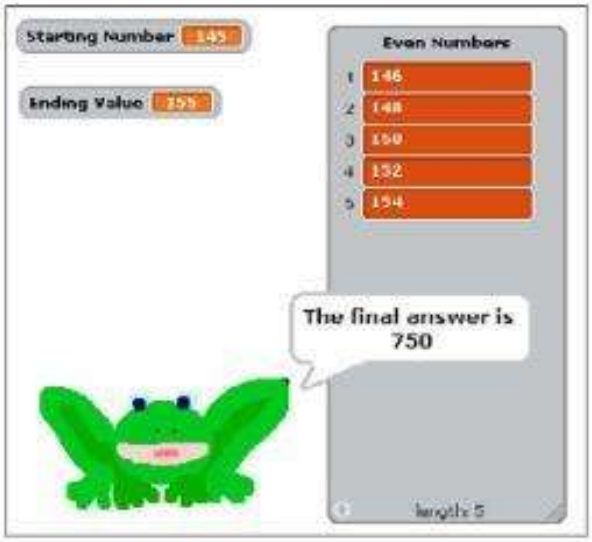
Remember that the list will only be displayed on the stage if a tick mark appears next to the name of the list. No **[hide]** or **[show]** instructions are available to display or hide the list.

It is useful to display the content of a list while you are writing and testing your program. Once you are sure your program works, you can hide the list.

2 Activity • Display even numbers and calculate the sum

Write a program to do the following:

- Let the sprite ask for the upper and lower limit between which even numbers must be determined and displayed.
- Store all these even numbers in a list (the list should be displayed on the stage).
- Calculate the sum of all the numbers and let the sprite display the answer.
- Output of the program can look as shown.



3 Activity • Low, high, average

A scientist receives a number of values as a result of an experiment that was conducted. A program is needed to read all these values and determine the highest, lowest and average of all the values. It also has to display how many values were larger than the average.

- Create a new Scratch program and a list (Values).
- Enter a number of values in a text file. Import the values from the text file to the Values list.
- Write code to do the necessary calculations and display the output.

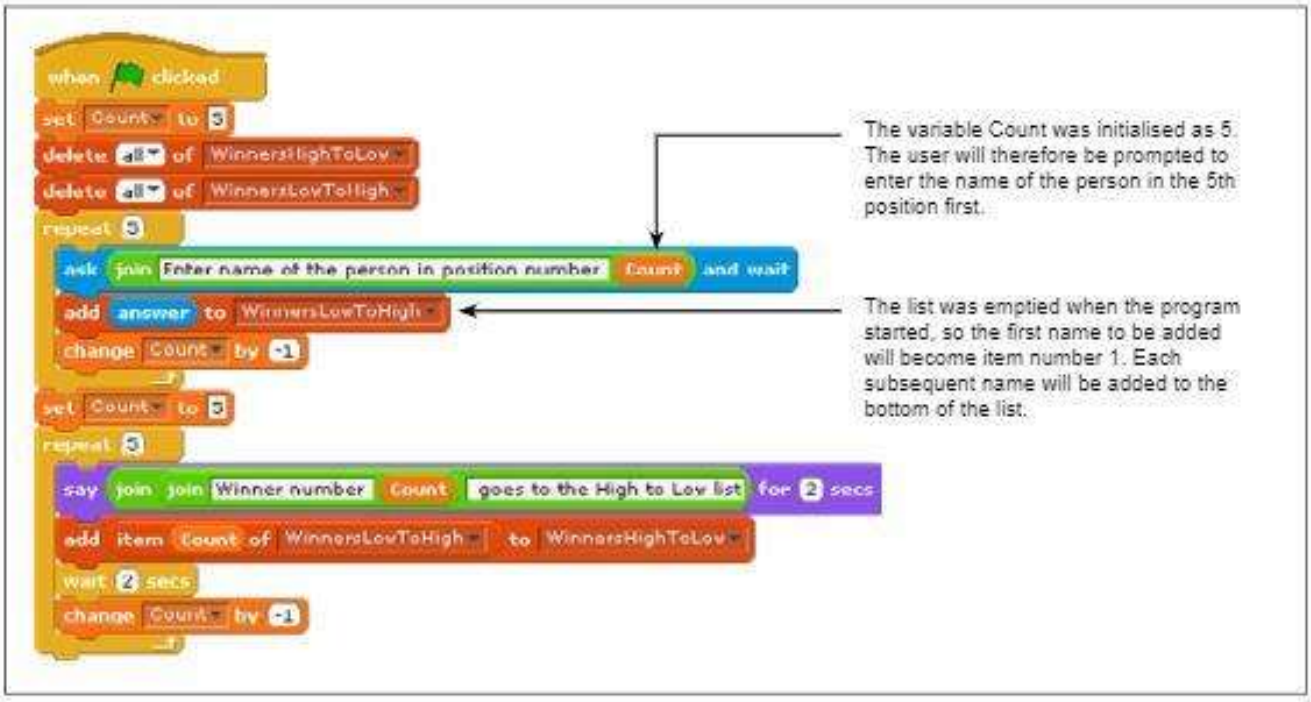
4 Activity • Find a specific item

Write a program to store a number of names in a list. (Decide for yourself in which way you want to put the names in the list.) Place sprites on the stage and write the necessary scripts so the user will be able to do the following:

- Display all the names that contain more than 10 characters.
- Display the first name in the list that starts with a certain letter. The user has to enter the letter.

Multiple lists

You can have more than one list in a program. The following program stores the list of names of five winners in a game in reversed order into another list. The names are entered in the list **WinnersLowToHigh**, starting with the person in 5th place. The winner's name is therefore stored in position 5 in the list. It then transfers the names to another list – **WinnersHighToLow**, so the winner's name appears in the 1st place. Note how both lists are cleared each time the program is executed by using the **[delete (all)]** instruction.



5 Activity • The start of a secret language

Create a program which will obtain 10 words from the user and store them in a list. Once the last word has been entered, each word must be reversed, and transferred to a second list.

6 Activity • Expand an existing program

Open the program **TestMathTables**. You need to extend this program. Here are some ideas on the way you can apply your knowledge and skills to add functionality to the program:

- Calculate and display the score and percentage.
- Add animation and sound to give an indication of whether the marks were good or bad (e.g. more than 80% people cheer and a sprite jumps up and down).
- Insert the correct answer in the place of each wrong one.
- Let the sprite ask which table the user wants to practice and not use random values.

7 Activity • Find and delete an item in a list

To celebrate Valentine's day, create a little game. Create a program containing two lists. One list for names of learners and another one containing a 'b' or a 'g' to indicate the gender of the learner. (Learner number 1's name will be in position 1 in the names list, and his/her gender in position 1 in the gender list.)

Use sprites to perform the following tasks.

- Use a sprite to ask and store the name and then the gender into each appropriate list. (Repeat this process to store any 10 different people.) Try to have an even match of boys and girls.
- Use another sprite to randomly match any boy with any girl. When you click on the sprite, he must randomly find the name of a boy, and then the name of a girl. He should then display ([say]) the two names next to each other.
  - *Challenge:* Modify the program so that once a boy or a girl has been used in a match, they will not be used in a second request for a match.
- Use another sprite to delete a specific name from the list of names. Remember to adjust the list containing the gender as well.

8 Activity • Extract information from list data

A class decides to have a fundraising through a raffle. You have to create an application which will keep track of the ticket sales. You will need to make use of two lists. Use one list for names and one for number of tickets sold.

Use sprites to do the following:

- Add a name and the number of tickets sold by the person to the appropriate lists. This application will allow for a person to have more than one entry. (Deliberately create duplicate names.)
- Display the total of tickets sold for a specific person from the list. The sprite should request the name of the person to be entered.
- Show the highest number of tickets sold as well as the name of the person who sold them.
- Display the average amount of tickets sold.
- Display the names of the learners who sold more than the average number of tickets.

9 Activity • Extract information from a list of data

You decide to create an application to display your report marks for the term. This application will require multiple lists. You need to create a list to store your subjects, and a list for the marks achieved in each of those subjects.

Use sprites to return the following results:

- Display your average mark for all your subjects.
- Display the name of the subject with the lowest mark.
- Create a new list to display the symbol achieved per subject. This list needs to be cleared and repopulated each time you click on this specific sprite.

10 Activity • Modify existing lists and populate additional lists

An application has already been written partially. This application is used to keep track of stock in your tuck shop. Open the program **TuckShop**. Take note of the following: The first list contains the product name, the second is the number of items in stock and the third indicates the recommended amount of items.

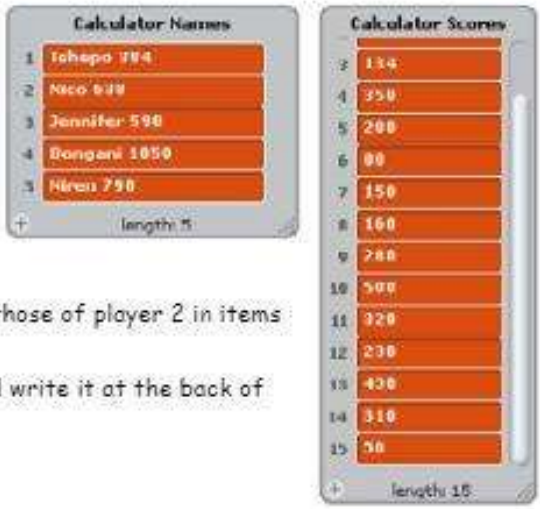
Use sprites to return the following results:

- Identify which products have less than 50% recommended stock.
- Generate a list of stock items less than the recommended level and add the word 'urgent' to the stock of which there is more than a 20% shortfall.

11 Activity • Calculate total score

A games program must store the names of players as well as 3 scores for each player. It must then calculate and display the total score (the sum of the 3 scores) of each player. Write a program for this part of the game.

- Create a list called Names. Store 5 names in the list.
- Create a list called Scores. Store 3 scores for each player in this list. (The scores of player number 1 must be stored in items 1 to 3, those of player 2 in items 4 to 6, etc.)
- Calculate the total of each player's 3 scores, and write it at the back of each player's name in the list Names.



12 Activity • Condense an address

In some address lists the addresses need to be as short as possible, but you must still be able to recognize the original address. One way to condense a sentence is to remove certain letters.

Write a program that will convert a number of addresses to shorter versions by removing the letters i, o, u and y.

This is an example of the output of such a program:

(Hint: Store the letters to be removed in a list, and use the **[contains]** instruction to test if a letter appears in the list.)

MODULE 1.7

Prevent and find errors

Data validation

In the computer world reference is often made to the *GIGO* principle (Garbage In, Garbage Out). This means that if the user types in garbage that does not match what your program is expecting, then your program is likely to either crash, hang, or produce invalid, incorrect, senseless output.

Data validation is simply the art of writing your program so that you check for and prevent a user error when data is entered. You can't check that the data is *correct*, but you can check that it is *valid*.

What's the difference between correct and valid?

Imagine you need to ask the user to enter their date of birth. There are several things they can do wrong here:

Scenario	Valid / Correct
A value is too small or too big for the day or the month. For example, there is no day 0, February does not have 31 days and there is no month 13.	Data outside of boundaries is **invalid**.
The user is asked to enter the date in the format dd/mm/yyyy, and it is entered as 12/04/13.	The data is **invalid**, because the format should have been 12/04/1914 or 12/4/2014.
Choose the current year as the year of birth.	This data can be **valid** if it is in the correct format, and has valid day and month numbers. It will also be **correct** if it is the birth date of a new born baby. However, it will not be correct for a person in matric or someone trying to register for a driver's licence.

1 Activity • Rules for validity

Give examples of valid values for each of the following:

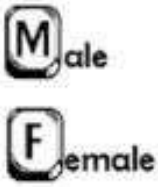
1. An ID number.
2. The grade a learner is in.
3. A cellphone number.
4. A person's name and surname.
5. An e-mail address.
6. A Twitter address.

Validating data that is entered by typing is a big issue in all programs. That is why most programs that use a GUI allow users to enter data by using lists, calendars and spin boxes. These are all graphic elements that only offer the users a limited range of choices and so help to prevent them from entering invalid data.

You can make use of a loop to validate data – and make the user keep on entering the data until what they type is valid and will work with your program. To avoid frustrating or irritating the user, you must make your instructions about what to input very clear.

You can also use sprites to create 'buttons' which the user should click to make a choice.

To give you an idea of what data validation entails, write the programs in the following activity. File the algorithms for all the programs – you will use it in programming a lot.

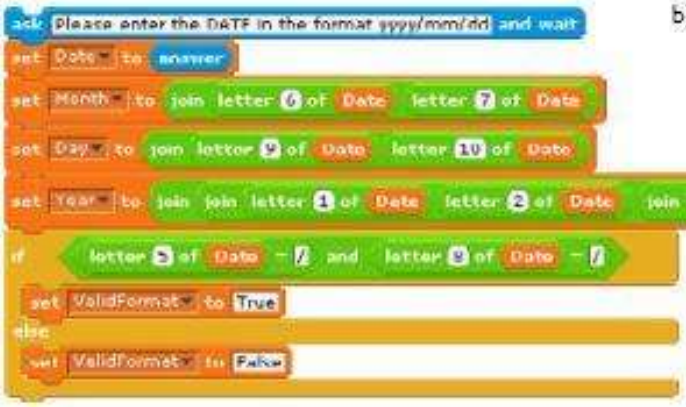


Use any of the costumes in the subfolders of the Letters folder, then add additional writing using the Paint Editor.

2 Activity • Data validation

1. The following program instructions demonstrate how the format of a date can be validated. Create a program containing the code, then add additional code for data validation as requested.

   a) Test if Month is a value between 1 and 12 (both included).



   b) Test if the day is between 1 and 30 for the relevant months, e.g. April, June, etc. (month values of 4, 6, etc.).

   c) Test if the day is between 1 and 31 for the relevant months, e.g. January , March, etc. (month values of 1, 3, etc.).

   d) Test if the day is between 1 and 28 or 29 for February. This one involves more work - you need to determine if the year is a leap year as well.

2. Write short scripts to validate each of the following. Display error messages which indicate clearly what mistake has been made in the input should the user enter an invalid value.

   a) Before you can calculate the square root of a number, you have to test if it is a positive value.

   b) Before you can do division, you need to ensure the denominator is not zero.

   c) The register classes in a school are named 8a, 8b, 8c, 8d, 9a, 9b,. . . . 12d. Write a script to read the name of a register class, and test if it is a valid class.

      Hint: Store the valid class names in a list, and use the **[contains]** instruction to test if the class that has been entered appears in the list. You can use this technique to test for any valid characters, numbers or strings.

   d) An ID number must contain 13 digits - no spaces, letters or other characters.

e) A cellphone number must comply to all the specifications for an ID number, but must consist of 10 digits. Try to write a program so the same code can be used to test for a valid cellphone number and a valid ID number.

f) The input (any value) must contain capital letters only.

g) The digits in the 7th to 10th position in an ID number indicate gender. A number between 5000 and 9999 indicates a person is male, a value smaller than 5000 indicates that a person is female.

Read the ID number and gender of a person. Test if the gender corresponds with the information in the ID number.

3. A company allows their users to create their own passwords for logging on to the computers. They do, however, specify rules that have to be followed when the password is created. These rules are as follows:
   - It must be between 8 and 10 characters long.
   - The first character must be a letter.
   - The second character must be one of the following: * ? /
   - The whole password cannot contain more than 4 letters - the rest must be numbers or other characters.

   Write a program that allows the user to enter a password, and then tells him/her if the password is valid. It should say exactly which rules have been disregarded if the password is not valid.

4. A college assigns an employee number to each staff member. It consists only of numerical digits. The last digit of the number is used as a check digit to validate the employee number as a whole. The last digit should be the remainder after the sum of all the previous digits of the number have been added and divided by 10. E.g.:

   Employee number 123455 would be valid because:

   1 + 2 + 3 + 4 + 5 = 15

   15 MOD 10 = 5

   Employee number 2365872 would be invalid because:

   2 + 3 + 6 + 5 + 8 + 7 = 31

   31 MOD 10 = 1 (to be valid the answer should have been 2)

Debugging your program

What is debugging? Well, strictly speaking, any part of a program that does not work as expected could have a 'bug'. There are a few tips and techniques that you can use for debugging any program that you write.

Use a trace table

Sometimes you are presented with an algorithm and are not sure what it does or how it does it. Luckily there is a tool that you can use, not only to help you understand what an algorithm does, but also to test it for problems when it does not seem to be working correctly. This tool is called the *trace table*.

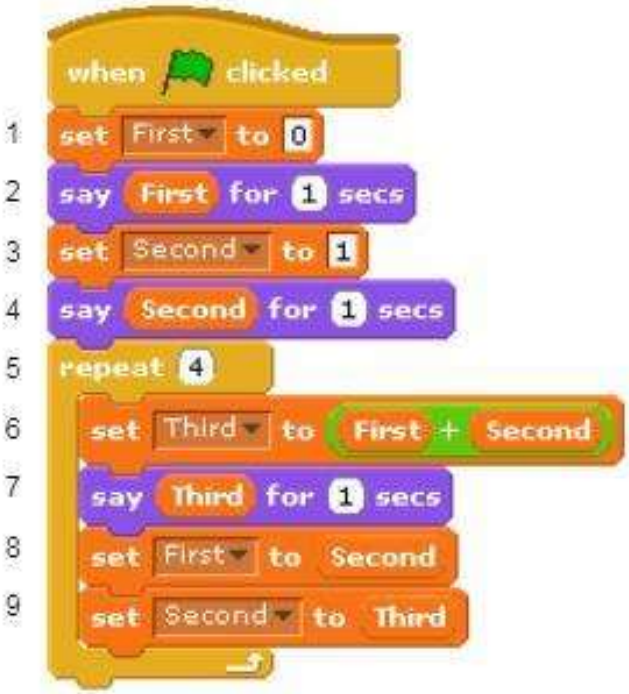
The trace table is a table which you use to follow the instructions in an algorithm (or even a printed program) step by step to see what it does.

ROTFL Cat says:

The steps you need to follow to create a trace table are:

- Create a table with columns for all the variables in the algorithm or code.
- Add columns for all comparisons/decisions.
- Add an extra column (the leftmost one) for the number of the instruction.
- Add an extra column (to the right) for the output.
- Add rows to the table as you go through the instructions – and change the values of the variables according to the instructions.
- Keep on adding rows until you reach the end of the algorithm or code.

Follow the example below to see how it is done.



Number	First	Second	Third	Repeat	Output
1	0				
2					0
3		1			
4					1
5				1st time	
6			1		
7					1
8	1				
9		1			
5				2nd time	
6			2		
7					2
8	1				
9		2			
5				3rd time	
6			3		
7					3
8	2				
9		3			
5				4th time	
6			5		
7					5
8	3				
9		5			

From the output it is now easy to see that the program produces the first 6 Fibonacci numbers. When you get to long and complicated algorithms, some people only do a trace table for the part of the algorithm that is not working as it should (and therefore only create columns for the variables that appear in that section).

Activity • Trace tables

The following programs do not do what they are supposed to do. Use a trace table to work through each program and determine what the logical error in the program is. Change the program so the output is correct.

1. Open the program **CountBackStuck** and execute it. The program is supposed to count backwards from 10 down to 1.

2. Open the program **CountBackInfinite** and execute it. The program is supposed to count backwards from the number you choose down to 1.

3. Open the program **ReverseName** and execute it. The program asks your name, and is supposed to write it in reverse.

Watching your variables

Most programming languages have a 'watch' feature that allows you to see the values of variables in a separate window while the program is running. Scratch allows you to see the value of variables on the stage. You either make sure that the variable has a 'tick' next to its name, or you can use the **[show]** and **[hide]** instructions to control when the variable is visible or not.

You can slow down the speed by making use of the 'Single stepping' function. Sometimes it helps to put a **[wait]** or **[say]** instruction in places in the program if you are unsure if the processing ever reaches a certain point.

Activity • Trap the cat

Open the program **CatEscape**. Use Single stepping or a **[wait]** block to find out why the cat is able to move out of the red frame.

MODULE 1.8

Projects

By now you have learnt enough basic programming techniques to design a good Practical Assessment Task (PAT). In this module we are going to discuss a few additional effects and techniques you can use to make a program more interesting. We are also going to give you some advice on planning a project and making it look more professional.

Special effects

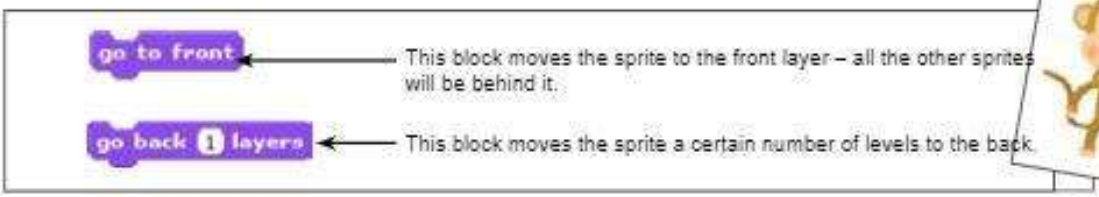
Let the background move

Sometimes the stage is just too small for everything that needs to happen. Maybe you want a much larger landscape as background, with a sprite that moves from one side to the other. To do this you can use more than one sprite instead of one of the backgrounds of the stage. This forms a kind of platform that moves from one side to another behind the sprite, so it looks as if the sprite travels a large distance. Such a program is stored in the **Examples/Games/ ScrollingDemo** folder.

This example shows how you can place a number of sprites next to one another so they are off the screen. The 'main character' sprite stays static, and you use a bit of mathematics to synchronise the 'landscape' sprites' movement so it seems as if the sprite moves.

Levels – move sprites forwards and backwards

Every sprite you place on the stage sits in its own layer. (If you have 3 sprites, there are 3 layers). When two or more sprites need to be placed on top of one another to create a certain scene, a sprite might not be in the correct layer. You can move a sprite one layer forwards or backwards using the **[go to front]** and**[go back ( ) layer]** blocks in the **Looks** instruction group. Look at the examples displayed here.



You can use these instructions in a creative manner to build a machine, for example, by using a number of sprites and placing them in levels on top of one another.

The importance of planning

You are ready to write bigger projects, but you should realize that complex code needs proper planning to make sure it works. The more impressive your program is, the more time you should spend planning what happens before you even try to create a single instruction.

Remember: Planning only really works if you record it – you need to draw diagrams, write down ideas and sequences, sketch out graphics, etc.

Some tips:

- If you are creating and animation then ***storyboard*** it.
- Break the program up into separate problems and create flowcharts for each of them:
  - What instructions are you going to give the user?
  - How will you receive input?
  - How will you present results?
- Work out how the whole program works together and create a flow diagram for that.
- Design your graphics and plan out their colours and interaction – keeping in mind the senses that you can use in Scratch whilst you do so.
- Plan for all the different types and combinations of data that the user might input and the mistakes you have to protect against (data validation).
- Plan where you will have to use
  - decisions
  - loops / repetition
  - counter variables
  - flags
  - broadcasting and receiving messages

  to solve the problems you are confronted with.

ROTFL Cat says:
The more you practice programming; the more new things you try; the more mistakes you make and fix – the better you become! This is the only real way to learn the art and skill of programming.

The final touches

What makes the difference between a good program and a great one? What makes the user really enjoy what you have created for them?

The answer is fine-tuning! You need to take the time and effort to make your program shine and gleam. The actual 'problem solving' part of the program normally takes less work than doing all the fiddly bits to get the user interface just right. Here are some examples of final touches:

- Instead of just 'saying' the results of a program, you could display graphics. For example, if you write a program where a coin is tossed, you display an image for the result instead of just saying 'heads' or 'tails'.
- Where possible, give the user interactive elements (buttons and other images) instead of making them type in data.
- Check the validity of data to prevent the program from freezing or giving invalid output.
- Use different sprites and backgrounds to add interest.
- Make sure that what you do is attractive and aesthetically pleasing!

The approach you need to take when writing a program is:

- Complete your code.
- Test your program:
  - Test that all the options work as expected.
  - Ensure that there are no unexpected results.
  - Make sure that your instructions are clear.
  - Make sure that the options mentioned in your instructions are available and work correctly.
  - Make sure that sprites reveal and hide themselves at the right times.
- Get others to use the program:
  - Watch them to see where they have problems.
  - Ask them for feedback.
- Run the program again and look for visual elements that 'don't work' or are jarring / not pleasing.

- Identify all the possible invalid data a user may enter. Enter it on purpose to see if the program can handle it.
- Go and make the changes that are required to solve the issues that have been identified.
- Repeat this process until the program is as close to perfect as you can manage!

Follow the design of a Coin Toss game

We will now design a game while following the approach in writing a program we discussed earlier. Work through this example, then apply these steps in creating you own projects.

Let's create a program in Scratch that allows you to have a coin flipping game. We make the game quite simple – the best of three tosses of the coin wins the game.

Describe the problem

The game is played by two players – the player in control, 'flipping' the coin (the program) and the player (the user of the program) who guesses which side of the coin will face upwards (head or tail) with the throw of the coin. It works as follows:

1. The player in control asks the other player to choose heads or tails.
2. The other player chooses.
3. The player in control tosses the coin into the air so that it spins.
4. The coin lands.
5. The player in control looks at which side lands face up and says either heads or tails depending on which was face up.
6. The player in control increases the count of wins for the person that has won.
7. Repeat this for as many rounds as the game is long (usually three).

To make this work on a computer you need to ask yourself questions about what is happening and then try to answer the questions using only the types of instructions that you know you can give the computer. An example of the types of questions and correctly-structured types of answers is shown below.

Questions and answers	Comment
What do we need to input? • choice for toss – heads or tails	We need to find out what data is input. Compile a list of all the input needed in your program and describe the types of data being input – and also what limits you expect on the data.
What do we need to count / store? • number of tosses • number of rounds played • number of wins • number of losses	Here is where you start making a list of data that you want to count / store. What you are actually doing is creating a list of variables that your program needs, that are not related directly to inputting data. These variables will hold the results of processing inside your program and are essential for making decisions and breaking long, complex calculations up into smaller steps that are easier to understand.
What do we need to display? • message with instructions • prompt for input of user choice • win or lose round • win or lose whole game	What you do here is work out all the information that your program needs to display. This includes messages that tell the user what to input, messages that give instructions, messages that report on errors and messages that display the results of the program. Whilst you are sorting out these, you should also start thinking about the best way to display these messages.

Questions and answers	Comment
Special features • display coin toss animation • display result of toss	In this section you allow yourself to dream and think about the fancy extra features that make your program nicer and better and more 'WOW' than another similar program. In this kind of program it might simply involve adding some types of animation to make your program more visually interesting.
Data validation • valid choice of heads or tails • repeat data capture until choice is either 'h' or 't'	This is where you look at the data that your program will be getting from the user and then, for every input, decide if a user error could cause your program to produce an error. If it could, then you need to work out the data validation you will need and plan to include it in your program.
Repetitions • animation - show start costume - repeat 10 times - show next costume - delay 0.1 seconds • whole round of game - ask for head or tail - check valid choice - display animation to simulate the toss - generate random number - if random number<50 then 'h' else 't' - if result = user choice then increase win else increase lose	In this section you plan the types of repetitions that you need in your program. Here you must think about what needs to be repeated and which type of repetition structure you need to use.

Activity • Planning the flow of the program

1. Create a flowchart based on the steps, and the questions and answers provided in the discussion above.
2. Create a basic test table that you can use for testing your program when it is finished.

Now let's look at translating all our preparation into a working program.

Creating the graphics

The graphics used for the program needs to be created. The following costumes will be needed for the sprites:

- **Sprite 1 – the computer contestant**

  This should represent the 'person' challenging the user to a game of tossing a coin... You can have just one costume or you can have multiples – a normal one for instructions, happy if they win, sad if they lose.

- **Sprite 2 – coin toss animation**

  Without an animation the toss itself (and so the program as a whole) is quite boring. You would probably need at least 10 costumes showing the stages of tossing a coin to make this work. 10 costumes with a 0.1 second delay between them = 1 second of animation.

- **Sprite 3 – result of the toss**

  You need at least two costumes for this sprite – a heads and a tails so you can show how the coin fell.

- **The stage**

  You can leave the stage as its default blank white background, or you can create more interesting backgrounds. The choice is yours. Remember that you can have more than one background for the stage.

2 Activity • Create the graphics

Create all the costumes, backgrounds and sprites that you will need for your program.

You may use the Scratch Paint Editor or any other image editor that you know how to use. For the best effect, save your images as .png format with a transparent background.

Tip: Your program doesn't necessarily need to toss a coin - in could do anything that captures your imagination. The more creative you are, the more the user will enjoy it.

NB: It is easy to get caught up in this step and spend too much time here. Beautiful pictures do not equal a working program and you need to deliver a working program to satisfy your customers (or get marks from your teacher). Be disciplined. Set a time limit and do not go over it whilst working on the graphics!

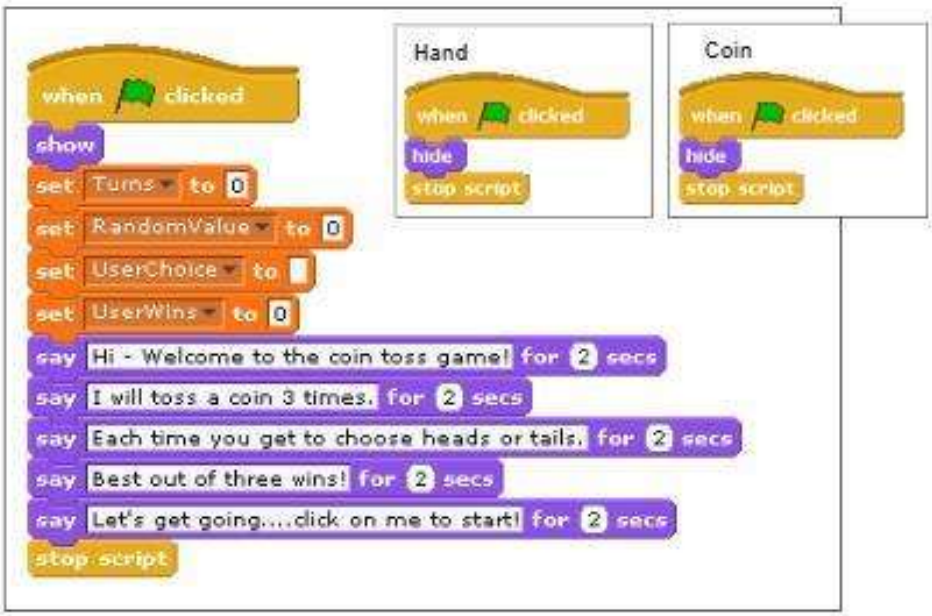
Setting the scene

Your program now has multiple sprites. You need to consider what should happen when the program starts. Are all the sprites meant to be visible? What stage are you using? Etc.

In this case the following probably applies:

- The main character is visible.
- All other sprites are hidden.
- You have set the stage to its starting background (not necessary if the stage stays blank).
- All variables are set to their starting values.

In our example, there are three sprites. ROTFL is the main character sprite, Hand is for the animation and Coin shows the result. The start-up scripts needed here are:



As you can see, only ROTFL has any serious instructions when the program starts. The other sprites simply hide themselves so that they are out of the way until they are needed.

NB: If the program needed the sprites to initialise variables etc., then this is the place to put those instructions.

Also: The script must have a 'stop script' instruction to prevent unpredictable behaviour in the program.

ROTFL's instructions:

- ROTFL is made visible (you can't assume that the sprite is visible).
- The variables used in the program are initialised.
- The user is greeted and instructions are provided.
- *NB:* You could provide instructions by having a different stage background with text on it as an alternative.

The main program loop

The ROTFL sprite now enters into the main loop that controls the program. Remember that, in our steps when we started planning, we said that the game needed to repeat the process of tossing a coin at least 3 times.

We could write the code out three times – but that would be very bad programming. Instead we use the programming structure of a loop to make our program much more efficient. We know that we need to repeat the tossing of the coin a fixed number of times, so we use the loop with a fixed number of repetitions to control the main flow of the program.

Now we need to follow the rest of the steps that describe the main flow of the program, namely:

1. The person in control (the computer) asks the other to choose heads or tails.
2. The second person chooses.
3. The person in control tosses the coin into the air so that it spins.
4. The coin lands.
5. The person in control looks at which side lands facing up, and says either 'heads' or 'tails', depending on which was facing up.
6. You increase the count of wins for the side that has won.

Increase the counter variable 'Turns' so that we can eventually end the program.

Ask the user for their choice and then use a **[repeat until]** loop to keep asking them to enter data until the first letter of what they type is either a 'h' or a 't'.

Store the choice that the user made.

Hide the ROTFL sprite and broadcast the message 'Flip' and wait until all responding sprites finish their scripts.

Now generate a random value between 1 – 100 (flip the coin).

If the value is less than 50 then

- broadcast the message 'Heads' to display a result
- wait for the responding sprite to finish its script
- show ROTFL sprite again
- If the user's choice = 'h' then say that they won the round and increase their score

If not then claim victory for the ROTFL sprite

If the value is NOT less than 50 then

- broadcast the message 'Tails' to display a result
- wait for the responding sprite to finish its script
- show ROTFL sprite again
- If the user's choice = 't' then say that they won the round and increase their score

If not then claim victory for ROTFL sprite

Let's take a look at how the other sprites respond to the messages broadcast by the ROTFL sprite.

Hand	Coin
	
The Hand sprite: • Plays a sound of flipping a coin • Switches to the first costume in the series • Shows itself • Repeats 10 times a wait of 0.1 seconds and a costume change • Finally it hides itself and stops the script	The Coin sprite: • Responds to both the Heads and the Tails messages. • Plays a sound • Changes to the appropriate costume • Plays a sound • Switches to the relevant costume • Shows itself • Displays text of which side is up • Hides itself and stops the script

Adding The final Touches

Finally, once the main loop is done, we should finish off the program neatly and tidily so that the user is left satisfied and happy. They need to know who has won the game overall. They also need to be given some sign that the program has finished, so they are not left wondering 'what happens next?'

The last part of ROTFL's script takes care of this. It is shown below:

- You will notice that we only counted the number of times that the user won. This is only a matter of personal preference in programming. You could have a separate variable to count the number of times ROTFL won if you wanted. That is up to your preferred style of programming.
- This code checks the number of rounds that the user won, and displays it along with a congratulatory message. Otherwise it displays a message showing that ROTFL won.
- Finally the program should have a 'Goodbye'-type of message before finally stopping the script.

We hope that you enjoyed this sample program and that you learnt a lot from it. Right now we have some challenges for you!

3 Activity • Adapt and improve

Create your own version of the program with some variations and improvements, e.g.

a) Add the 'Goodbye' and the stop script.

b) Allow the user to choose the number of rounds (more than two, fewer than eleven).

c) Stop the game as soon as someone wins (instead of playing all the rounds).
Hint: Someone wins when they have won more than half the rounds in the game.

d) Add suitable backgrounds

Flags and interactive elements

Now let's look at how we can create the same program with interactive elements to remove the need for the user to type in their choice of heads or tails.

What we need to do here is get rid of the text input dialog box and let the user click on a picture instead. That way they can choose heads or tails without having to type anything.

Using interactive elements in this way removes the need to do data validation (they can't type in any other random word) but creates problems of a new sort. Think about this:

- **Only sprites can respond to mouse clicks.**
  This means that to give two choices you need two sprites – and more if there are more choices. You have to manage how and when these sprites appear and disappear, and what happens when the user interacts with them.
- **You need to manage the appearance and disappearance of the sprite 'buttons'.**
  This has to happen if the program is to work properly. The problem is that only one sprite gets to respond to the mouse click – and then you need to make the others disappear as well. That's where smart use of a 'Flag' variable works wonders. We'll talk about 'Flags' in a moment.

These are just two problems that are applicable to the Coin Toss program, but you should be able to imagine that, when you have a whole lot of interactive control elements on the screen, managing them can be quite difficult to keep track of.

What is a flag?

A flag is usually a Boolean variable with stage scope, and it is used to let different parts of your program know what the state of the program is.

What are the interactive elements in this program?

In the Coin Toss program we want to display two sprites for the user to click on to make the choice for heads and tails, instead of typing a letter. We want these sprites to disappear after the user has clicked on any one of them.

How is the flag used in this program?

The program will have to wait until the user clicks on a coin (head or tails). A flag will be used to indicate when a coin has been selected. The flag will be a variable on the stage called ChoiceMade. This is how we'll use this flag variable:

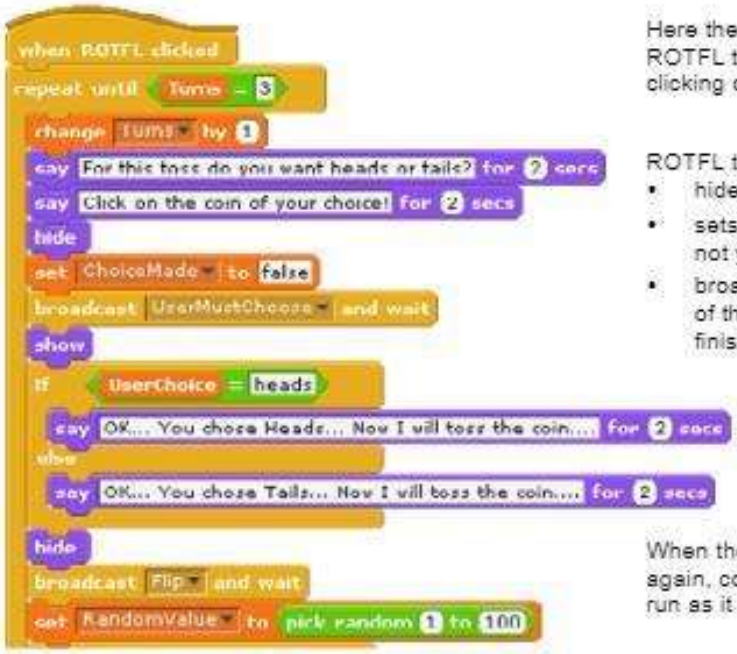
- This variable will hold FALSE when waiting for the user to make a choice.
- All the coin sprites will show themselves then check for a mouse click until the value in ChoiceMade changes to TRUE.
- When the user does click on a coin, one of the instructions that need to be executed will change the value in ChoiceMade to TRUE.

When the value stored in ChoiceMade changes, it is like a signal flare to other parts of the program that something has happened, and they can respond in the correct way.

Let us look at the complete program so you can see how the use of a flag and interactive elements change the program.

Take a look at how the two sprites (Heads and Tails) uses the flag **UserChoice** to control the program:

A program may have many flags, and their purpose is to let different parts of the program share important changes in what has happened in the program.



Here the program changes from the previous version – ROTFL tells the user that they will make their choice by clicking on one of two coins.

ROTFL then does the following:

- hides itself
- sets the flag variable to show that the user has not yet chosen anything
- broadcasts a message and waits for the scripts of the sprites who react on the messages to finish (we will look at those scripts shortly)

When the user has made a choice, ROTFL appears again, confirms the choice and then lets the program run as it did before.

Either way of writing the program (with text entries or with interactive elements) is correct. What really matters is what your customer wants – or what you think will work better for users.

The sprite receives the message and . . .

- shows itself
- starts a loop (**[wait until]**) that ends as soon as the flag variable ChoiceMade is set to TRUE
- sets the flag (ChoiceMade) to TRUE when the user clicks on the sprite, and the variable UseChoice to 'heads' or 'tails', depending on which sprite was clicked

The following activities contain suggestions and helpful hints to create more complex games using many of the techniques and effects discussed in the previous modules.

4 Activity • Create a race track

1. • Create a racing car sprite.
   • Create a **[forever]** loop and scripts that do the following:
      - If the left arrow is pressed, the sprite turns 2 degrees to the left.
      - If the right arrow is pressed, the sprite turns 2 degrees to the right.
      - If the down arrow is pressed, the sprite moves 5 steps backwards.
      - If the up arrow is pressed, the sprite moves 5 steps forwards.
   • If the sprite touches the edge of the stage, it should bounce.
   • Create a track on the stage:
      - Make the edge of the stage a specific colour.
      - Prevent the car from going off the track by detecting the colour at the edge of the track and bouncing.
      - Make sure it is possible to drive all the way around the track.

   Test the game to make sure it works right.

2. • Create another different racing car sprite.
   • Add different control keys to control this sprite's movement (e.g. the keys<w>; <a>; <d>; <x>).
   • Add the same instructions as for the first car so this sprite can't go off the track either.
   • Change the forward movement for each sprite to be a random amount.
   • Make the sprites bounce off each other if they touch.
   • Add a finish line on the track and let both sprites detect it if they touch the finish line.

   There you have just created a two player racing game!

5 Activity • Create an obstacle game

- Create a sprite that the user controls using the keyboard.
- Draw either a background or other sprites that the user needs to dodge as an obstacle course (you can create rivers, trees, fiery obstacles, cliffs - anything that takes your fancy).
- Add sprites that will give the user special abilities (cross fire or cross). If the user sprite touches one of these special sprites it gains those abilities to overcome obstacles.
- The aim of the game is that the user should navigate to a target on the screen and survive.
- Use two lists to allow you to choose different messages at random whenever something goes either wrong or right - so that the game is more interesting...
- Make it more difficult (for the user and for yourself as programmer):
   - You can use the (timer) to add a time limit.
   - Add a limited number of 'lives' that the user can lose as he/she makes mistakes.
   - Use the (timer) to animate some of the obstacle sprites- and make them move around.
   - Keep track of the time and use a list to record high scores.

6 Activity • Analyse a large program

Open the program **BushveldRacing**. This program uses some of the techniques learnt in all of the previous modules.

A huge part of your skill as a programmer involves being able to read and understand programs written by other programmers.

Your task in this activity is to read and use the program and then

- list all variables and their datatypes
- create a flowchart of the program's operation.

Section 2

Systems Technologies

Modules

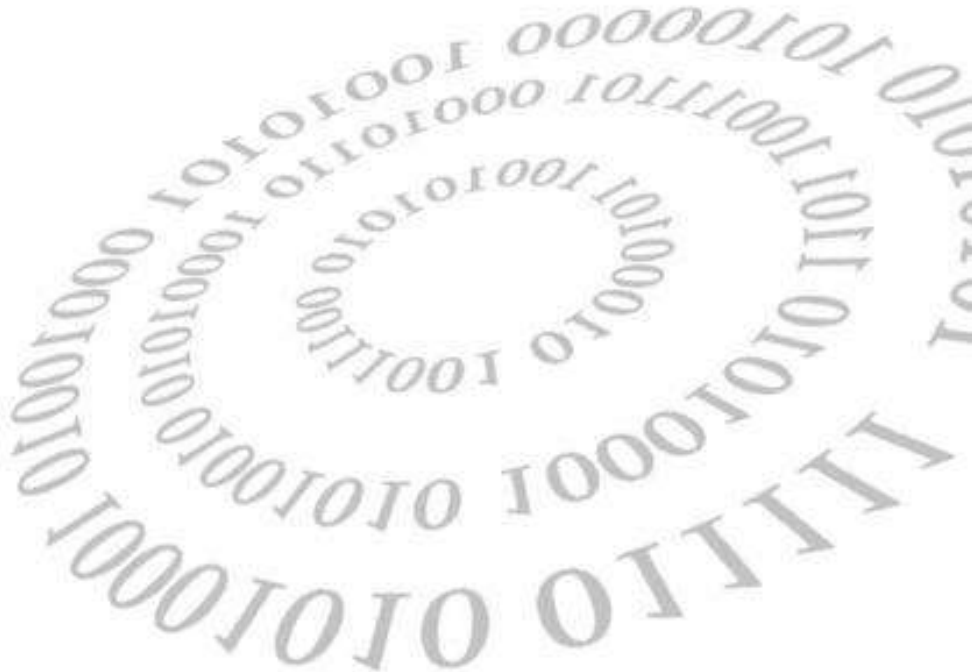
Module 2.1 Basic concepts of computing

Module 2.2 Basic concepts of hardware

Module 2.3 Basic concepts of system software

Module 2.4 Computer management

Module 2.5 Improve your browser



MODULE 2.1

Basic Concepts of Computing

What is a Computer?

The term IT (information technology) refers to a very broad and continuously expanding field to define or study. We need to start somewhere, however, and seeing as the computer (or computing ability) underlies almost every aspect of IT we will start by gaining an understanding of what a computer is.

Main Components or parts of a Computer

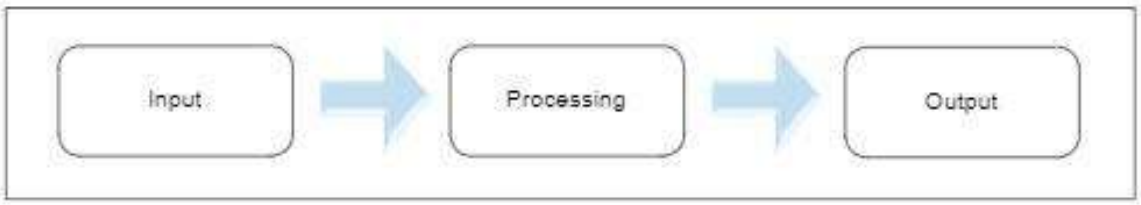
Any computer has two main parts, namely *hardware* and *software*.

- **Hardware** refers to all the parts of a computer that you can physically touch – the screen, the keyboard, all the cables and components inside the computer, etc.
- **Software** refers to the programs – sets of instructions – that tell the computer what to do (down to the smallest detail).

Without software the hardware would be useless and vice versa.

The basic model of a Computer

A computer is a ***system*** because it is an arrangement of many parts that work together. The parts are hardware (devices) as well as software (programs). All computer systems follow the same basic model, namely:



The fact is that most aspects of modern life work as systems. If you train yourself to understand what is 'fed' into the system (input), what processing is done and what output is delivered in all of these systems then you will start to see the world in ways that make it possible for you to computerise these systems! We refer to this model as the Input-Processing-Output (IPO) model.

A definition of a Computer

A computer is a multipurpose electronic tool that can receive data (input), can process the data, can produce results and can output them.

Computers can do this because someone has loaded prewritten instructions (a program or programs) which tells it exactly what must be done.

Data is raw, *unprocessed facts.*

Information *is the result of the processing of data and should be useful and meaningful.*

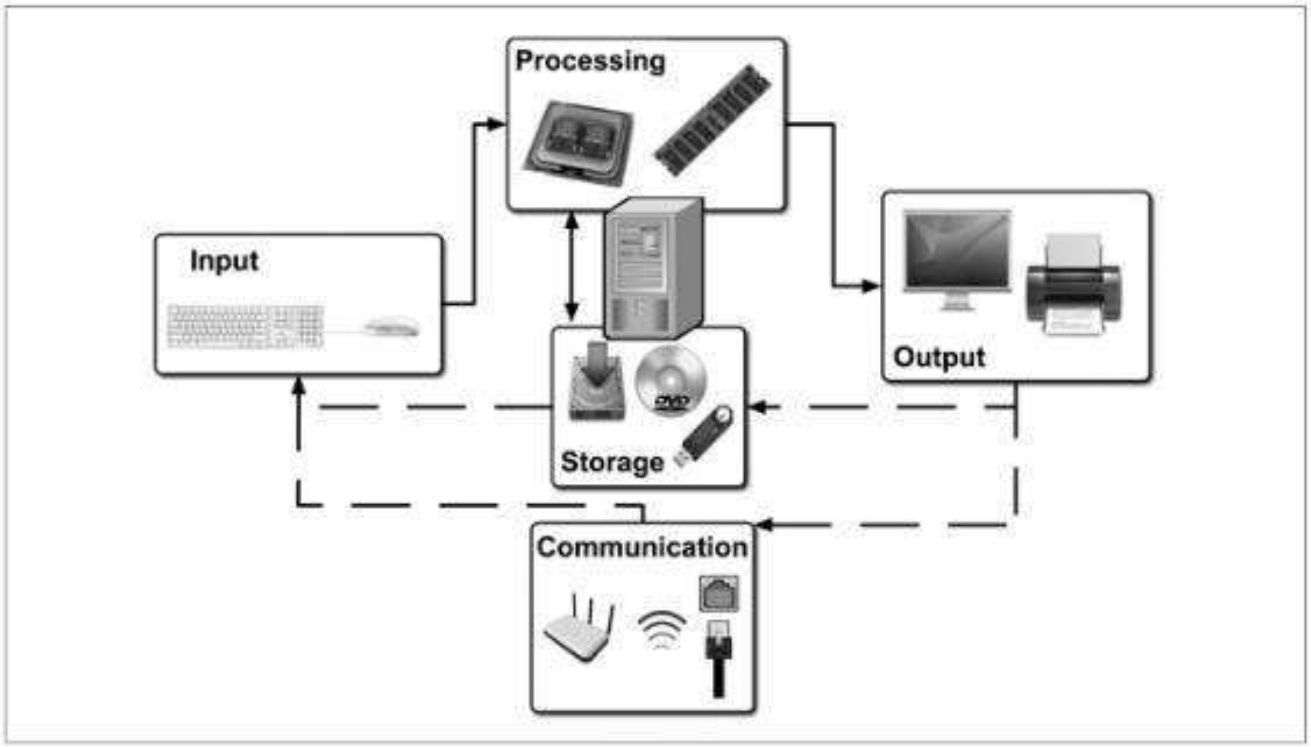
For example, the letters and numbers CAT123NW can represent some data. However, when it is processed or interpreted, we can see it as *information* in the form of a registration number of a car from the North West province.

Dedicated devices versus general purpose computers.

Note that, when we talk about computers, we refer to 'general-purpose' computers that can perform more than one function. For example, traffic lights and modern washing machines also follow the IPO model, but are designed to perform one specific task only. In a similar vein, modern cell phones have evolved from dedicated devices (used to only make and receive phone calls) to so-called smartphones that perform a variety of tasks. These tasks are all made possible by having different types of programs (software) loaded on the respective devices.

An expanded model of a Computer

The IPO model of a computer is actually a bit over-simplified. What we want to add to the computer model is some way to *store* data/information/instructions and some way to *communicate* these to other computers. Both storage and communication can act as forms of input or possible channels for output – but it is important that we think of them as separate parts of the system. So, the expanded model looks as follows:



Different hardware devices form part of the input, output, storage and communication components.

- **Input devices** allow us to get data into the computer. The most common input devices are the keyboard and the mouse.
- **Output devices** allow us to get feedback from the computer. The most common output devices are the screen, the printer and the speaker.
- **Storage devices** allow the computer to keep instructions and data that it is not currently working on, so that they can be accessed and used when needed. The most common storage devices are the hard drive, CD, DVD and flash disks.
- The **processor** or Central Processing Unit (CPU) is the part of the computer that executes the instructions (the software), processes the data and manages and controls all the other parts of the computer. Intel is a large manufacturer of CPUs for computers.
- **Memory** is the place where the computer temporarily stores those sets of instructions and data it is currently working with. Programs and data cannot be processed unless they are temporarily loaded into memory.
- We also commonly refer to this memory as RAM (Random Access Memory).
- **Communication devices** allow computers to communicate with each other. Equipment used for this includes routers, modems and switches – but we will learn more about these later.

Note the following:

- Programs have to be 'loaded' – transferred from storage to memory – before they can be used.
- Data must also be in memory if the computer is to work with it – the data can come from user input or from storage.
- The contents of memory only become 'permanent' when you 'Save' them – i.e. transfer them to storage (memory is erased when you close the program / switch off the computer – if you did not 'save' the contents of memory is lost forever!)

The software provides the instructions that make it possible for the computer to carry out tasks. These instructions take the input and put it into memory, process the data in memory, display output on the screen, etc.

1 Activity

1. What is the difference between hardware and software? Refer to suitable examples in your explanation.
2. How does a computer 'know what to do' to perform a task?
3. Briefly explain the difference between data and information by referring to suitable examples.
4. Describe what we mean when say we can use the IPO model as a simple model for a computer system.
5. Give two examples of input devices and two examples of output devices.
6. Briefly describe the role of the processor or CPU in a computer.
7. Why do we need both memory and storage in a computer system? Explain your answer by referring to the different roles that these two components play in a computer system.
8. Will you lose all the data and programs on your computer if the power goes off? Explain your answer.
9. Many electronic devices are actually computers, and all computers follow the same IPO model. Now classify the following aspects of different computers as input, processing or output:
   a) Touching an icon on the screen of a smartphone.
   b) The music you listen to on your MP3 player.
   c) The voice on your GPS giving you directions.
   d) Entering your PIN into an ATM.
   e) Using the remote control for your satellite decoder.
   f) The slip you get at the till in a supermarket.
   g) Sorting a list of names alphabetically.
   h) Calculating the VAT on a purchase.
   i) Receiving an SMS.
10. A computer can be defined as a multi purpose electronic tool that can receive data (input), can process the data, can produce results and can output them.
    a) Using examples, briefly explain, why we can say a computer is a multi purpose electronic tool.
    b) What enables a computer to be multi-purpose?
    c) Most schools have some kind of computer program into which teachers enter the learners' marks. Give an example of data that could be entered, and a type of output that could be generated.

The role of software

We explained before that all computer systems consist of both hardware and software. Software manages and controls hardware. You have seen that hardware is divided into different categories, according to the work that it does. Software, however, can be categorised in more than one way.

Software categories according to function

These categories describe the work that the software does. The two categories are:

System software is the software used to manage and control the operation of the computer. You cannot use a computer (or any type of IT device) without some form of system software to manage the way the computer or a device works. The most commonly referred to example of system software is the operating system. Examples are Microsoft Windows, Linux, Android and the Apple OS X.

Application software, on the other hand, is a whole set of additional programs that allow you to use the computer to do many different things, for example, to write letters, do accounts and budgets, play games, design houses and cars, make movies, etc.

Here are a few examples of application software packages that are commonly used:

Type of program	What it is used for	Examples
Word processor	Used to create, edit, format and print text-based documents.	Microsoft Word OpenOffice Writer
Spreadsheet	Allows you to perform tasks that are based on calculations. Provides charting facilities.	Microsoft Excel OpenOffice Calc
Databasis	Can be used to store data in an organised manner, manipulate the data and process the data into information.	Microsoft Access OpenOffice Base MySQL
Presentation software	Can combine text, sound and graphics, including animation, into a presentation – usually in the form of an electronic slideshow.	Microsoft PowerPoint OpenOffice Impress
Graphics and design	Used to create and manipulate images (graphics) on the screen.	CorelDraw, Adobe Photoshop, Adobe Illustrator, Macromedia FreeHand, AutoCAD and Microsoft Paint.
Communication software	Allows the user to communicate electronically with other users on a network.	Microsoft Mail and Outlook (e-mail) Pegasus Mail Skype and Google Chat for online chatting and conferencing.
Web browser	A program that is used to access and view web pages on the Internet.	Internet Explorer, Opera, Safari, Mozilla Firefox and Google Chrome
Games	Most modern computer games allow you to enact roles and have adventures in virtual worlds or scenarios. Many of them provide interfaces which make use of very realistic 3D graphics and sound effects.	Grand Theft Auto World of Warcraft
Web authoring software	Used in the creation and design of websites.	Microsoft Expression Web, Microsoft SharePoint Designer and Adobe Dreamweaver
Plug-ins	Special plug-in programs add extra features to software – e.g. in a web browser they are needed to allow the user to view 'special' effects. This has become necessary as more and more multimedia (a combination of text, pictures, sound and video) is added to the web.	Flash QuickTime

Software categories according to distribution models

Knowing and understanding the different ways that software is distributed is important, because it allows you to make good decisions about the software that you want to use and install on your computer. Each type of software distribution has its advantages and disadvantages relating to

- Cost
- Ownership
- The right to make changes and improvements to the software

Open source software (OSS)

Open source software is software made available with a license that allows you to access and modify the source code, but not to sell the software – and the changes you make go back to the community that uses the software.

OSS is therefore more than simply free software. It is software that is not only freely available, but of which you can also get the *source code* to the software. Any changes to the software that you want to see happen, you can make yourself. In other words, you do not have to wait for a company to add a feature or fix a bug – you can do so yourself.

Whereas OSS refers to a specific distribution model, *Free Open Source Software* (FOSS) is more of a general social and political movement by organisations such as the Free Software Foundation (FSF). The aim of using the word 'free' in the name FOSS is meant to place emphasis on the freedom of the user to modify, copy and re-use the software – it does not refer to the 'price' at all.

Advantages

- No cost / low cost (usually free).
- Access to source code.
- Freedom to adapt and change the software to suit your own needs.
- Access to a community of users / support.

Source code is the programming code used to develop a specific application.

Disadvantages

- No one to hold responsible for glitches and shortcomings in the software.
- Risk of poor support.
- Higher skills needed if you want to use this model to its fullest potential.
- Various 'varieties' of the same software can arise.

Linux is open source software (OSS) but I can buy it in a box at my software store – what is going on?

The OSS license prohibits you to sell OSS. When you buy Linux or any other OSS in a box, you are paying for additional utilities and tools (like a nice automated install system) and extra help and manuals – and for support (a number to call for help). You will still get the full source code of the software (and usually you are allowed – even encouraged – to give the software to friends and family to install it on as many computers as possible!). The OSS business model allows companies to make money by selling services and support for the software.

Proprietary software

Proprietary software is owned by a company – you never get the source code – in fact you only ever 'license' the software, you do not 'own' the software. The licence outlines the conditions under which you may install and use the software.

Proprietary software usually involves a cost (there are exceptions) – in fact it can be expensive and you cannot add features that you want or need or even to fix a simple bug that irritates

you. What you pay for is the privilege of having someone to hold responsible should the software not work the way it is supposed to. Note, however, that most proprietary software's license terms limits the providers liability – and you cannot use the software without accepting the license terms.

Proprietary software uses three main distribution models, namely:

Shrinkwrap	Shareware	Freeware
What you buy in stores, software in boxes sealed with shrink wrap.	Download and copy/share with your friends.	Download and copy/share with your friends.
• May not be copied. • May generally only be used on one computer.	• Usually limited in features/time. • Uses the 'Try before you buy' concept – if you like the software and want to use it past the trial date or access all the features, you will need to pay a license fee. • Usually cheaper than shrink wrapped software because you do not have to pay a 'middle man'.	Differs from open source software, because the software is free, but you cannot access the source code.

2 Activity

1. Name and briefly describe the function of the two main software categories.
2. Give the generic name we use to describe software that is specifically designed to perform tasks that are based on calculations and provide charting facilities.
3. Briefly describe what database software is used for and give two common examples of this type of software.
4. Would Microsoft Word be an example of system software or application software? Briefly motivate your answer.
5. Give the generic name we use to describe software used in the creation and design of websites.
6. What is the difference between presentation software such as PowerPoint and graphics and design software such as Adobe Photoshop?
7. Give the generic name we use to describe software such as Internet Explorer, Opera, Safari, Mozilla Firefox and Google Chrome.
8. Why do we need plug-in software such as Flash and Quick Time?
9. What is the main difference between open-source and proprietary software?
10. Give two potential advantages and two potential disadvantages to using open source as opposed to proprietary software.
11. What is the difference between freeware and open source software?
12. What is shareware?

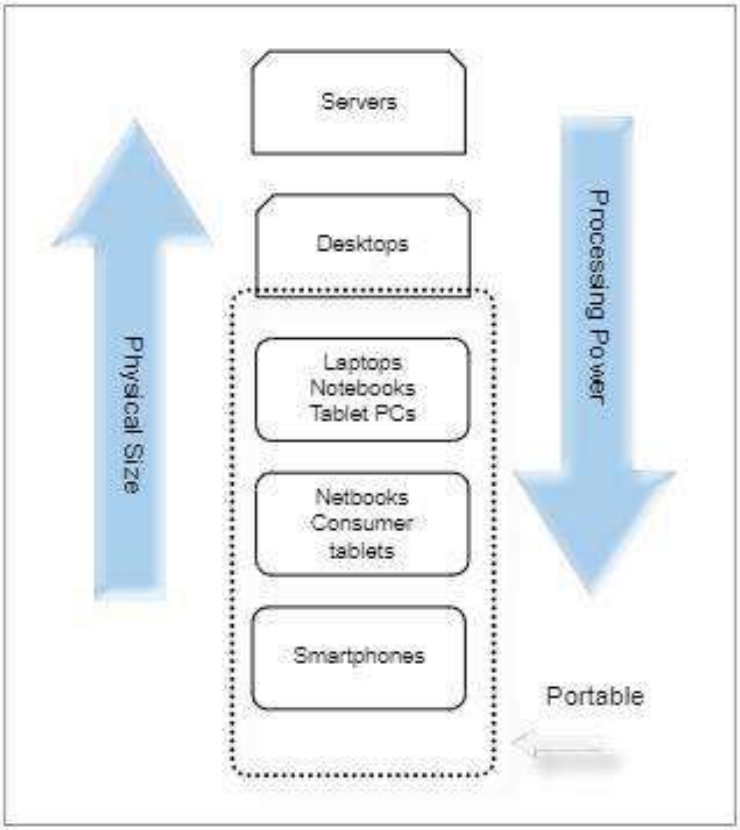
Types of Computers

You should have already realised that computers are everywhere and take many different forms. We are going to explore the more obvious types of computers that you will encounter.

Size and processing power

Some computers are more 'powerful' than others. By this we mean that they can process more data and run programs and retrieve information more quickly. We all obviously want the most powerful machine (we can afford). 'Powerful' in computer terms is generally a combination of

- the speed and performance of the CPU(s)
- the amount of storage
- the size of memory, because the larger the memory is, the more programs and data can be loaded to be processed at any one time.



Please note that this diagram gives a very general hierarchy in terms of computing power and physical size. (E.g. some laptops are more powerful than some desktops.)

Desktop

Laptop

Netbook

Remarks:

- Servers are powerful computers that are used in network environments to help 'run' a network of linked computers. They are not used by users directly, but are used to supply services to connected computers and users on the network.
- The term 'Tablet' is used for two completely different categories of devices, namely consumer tablets and Tablet PCs. The difference between these categories is summarised below:

- Consumer Tablets

  These are small portable devices like the iPad, Galaxy Tab, Playbook, Xoom and HP Touchpad.

  They are designed to be controlled through touch screen interfaces, but you can get a 'pen' or 'stylus' to use with them. Such a 'pens' is actually nothing more than a substitute for a finger – i.e. It does not increase accuracy or work like a proper pen input device.

  Consumer tablets usually have WiFi, Bluetooth and 3G communication built in.

- Tablet PCs


These are normal laptop-style computers (with or without keyboards) that have the ability to work with a pen-like stylus for pen-based input. This pen is a precision device which allows for accurate and pressure sensitive input and is perfect for applications such as digital art or digital handwriting.

They are much more expensive than consumer tablets and have a shorter battery life. They are also more expensive than normal laptops and so have not sold well.

- We referred earlier to the difference between general purpose computers and dedicated devices. Dedicated devices are also known as embedded systems. These devices have all the elements of a computer from an IPO perspective but cannot be programmed or expanded. Examples are microwaves, GPS, MP3, hospital equipment and ABS brakes.

Uses

When purchasing a computer, we also need to look at what we intend to use the computer for and whether we want to take it with us (whether it needs to be portable). It is becoming increasingly difficult to provide a clear distinction between the capabilities of all the different types of computers, as they tend to overlap more and more these days. For example, it won't be too long before our cellphones are performing the same functions at the speed of our 'normal' computers.

Type of computer	Typical use
Server	Providing services to users in a network.
Desktop, laptop, notebook, netbook, Tablet PC	Productive work and the use of applications that allow one to be creative.
Consumer tablet	Use digital content, access the web and e-mail, create documents, play games, etc.
Smartphone	Mobile access to connectivity and services.

Advantages and disadvantages of using Computers

If computers are so wonderful, why is everything in the world not done by computer?

Let's first look at why we do use computers: Computers

- are fast and accurate
- are good at repetitive tasks
- do not get tired or complain or demand increases in pay
- are multi-purpose
- enable faster communication
- are excellent at keeping records
- allow for many and various methods to process and analyse data.

So what are the major *disadvantages* in using computers? Computers

- are only as good as the way that they are used and the data that is fed to them
- can only be used for a limited number of physical applications (i.e. need machines and robots to be able to manipulate physical objects)
- can be very expensive
- can not think or solve problems
- need trained and skilled programmers and operators
- might need specially developed software to solve a problem (expensive)
- quickly become outdated.

3 Activity

1. Some types of computers are generally more powerful than others. What makes one computer more powerful than another?
2. Why would an end-user not generally buy a server? Explain your answer by referring to the typical role that servers perform.
3. List three types of portable computers.
4. Arrange the following in descending order of computing power: Server, Smartphone, Laptop, Netbook.
5. a) Name three consumer tablets.
   b) What are these tablets typically used for?
6. List three benefits of computers. Motivate your answers by referring to scenarios where each of the benefits is applicable.

ICT systems

We live in what is often called the 'Information Age'. In many ways the ability to collect, process and send information quickly has become even more valuable than physical resources such as gold.

For example, if you run a supermarket, and can tell which products people like best, and when they prefer to buy them (e.g. winter clothes in March/April and summer clothes in September/October), then you can maximise your profit by ensuring that you supply them in time.

The ability to analyse and interpret data in order to arrive at valid conclusions and to form accurate predictions (information) can spell the difference between success and failure.

Did you know, for example,

- that newspapers and magazines make almost as much (if not more) money from selling the information about their subscribers as they do from selling their actual product?
- that Google and Facebook both make their money through advertising? By collecting information about their users, they can sell advertisers the ability to show you advertisements for products that you are likely to be interested in.
- that using information to streamline their stock control and ordering processes can double the profitability of a business?

Information is power and creates the potential for wealth.

What is an ICT system?

Technology on its own is useless. We need to combine technology with people who manage and integrate it into a system that focuses on getting a particular task done efficiently. This is where we begin to talk about IT/ICT *systems*.

An ICT system is a combination of *hardware*, *software*, *data*, *processes* and *people* with the purpose to collect, manipulate, convey, store and retrieve data and information.

Information	Manipulated or processed data
Communication	The process of transferring data/information from one place to another
Technology	• Systems technologies (hardware and software) • Communications technologies (networks and communication devices) • Internet technologies (Internet, World Wide Web and e-communication)

The fact is that an ICT system is a *system* which functions just like every computer system. It has the same main components (input, processing, output, storage and communication) – it just operates on a large scale.

ICT systems can be relatively small (e.g. a system that is developed for and is used to run your local video rental shop that is NOT part of a big chain of stores) or they can be huge like the systems used to run hospitals or supermarket chains or communications companies. Large chains need efficient and speedy data communications to link all the separate locations together and to gather the data from each store to a central point for efficient management and planning.

The amount of data that businesses store is huge. There are two complete and complex fields of study devoted to working with it –

- *Data warehousing* – how to securely store, manage and retrieve such large amounts of data.
- *Data mining* – how to process and analyse this data looking for valuable information that you did not even know was there.

Remember

ICT systems also include the human aspect – that means that procedures and paperwork must be developed to ensure that the right type of data is captured and reports are correctly generated – and people must also be trained to use the system effectively. Developing an ICT system is much more than just coding a program!

Examples of ICT systems in everyday life

We are going to take a brief look at two examples of ICT systems that you see in use almost every day.

Supermarket chain

Supermarket chains have many stores in locations all around the country. In a supermarket chain an ICT system will be concerned with managing the movement of stock as efficiently as possible so as to maximise profitability and customer satisfaction. The category of software developed for this is technically a POS (Point of Sale) system, but this has typically grown into a much larger system that manages stock control, finance, human resources (payroll), etc. Here's a brief summary of some aspects that such a system deals with to make the business work better:

- Modern POS systems use barcodes and scanning to make the process of adding up the total bill faster and more efficient.
- Scanning is also more accurate and prevents errors in the capture of prices that occur when people type in the prices.
- This also allows the items scanned (because you now have detailed information of what is going out of the shop) to be removed from a computerised database of what is in stock – enabling you to better manage your inventory.
- The data is sent to the head office so that stock ordering and management decisions can be made efficiently for the whole group.

An ICT system enables them to

- order only the right amount of stock when they need it
- find out which items are not selling well and remove them from the shelves
- track sales so they know what sells well in different stores, and make sure those stores cater better to their customers.

After a few years you build up a detailed database of sales and can analyse them to see trends of when people buy what, so you can increase and decrease stock of items when people are more or less likely to buy them (for example Easter eggs during April).

If your customers use loyalty cards or pay electronically you can track their spending habits and the stuff that they are interested in. You then sell this information to marketers who are looking for that type of customer.

Cellphone company

A cellphone company needs complex ICT systems to manage a whole range of items and processes such as:

- contracts
- ensuring adequate supplies and stock of cellphones
- generating finance for expansion and maintenance
- tracking (and billing) of call durations, SMSs, data transfers, free talk time, carry-over of free minutes/seconds, and expiry of the same
- service and repairs of cellphones and other infrastructure
- monitoring the network status
- maintaining staff records

A cellphone company is an example of a company whose business is almost all about data and information. Data is collected about every phone call or data transfer that you make. The company can then

- list how many calls you have made, to which numbers and how long each call was
- total the number of minutes used and bill you for them
- total the amount of data you have transferred (usually in megabytes) and then bill you for that.

Activity

1. ICT stand for Information Communication Technology. Briefly describe what each these terms refer to.
2. Information is now regarded as a valuable commodity. Explain what this means by referring to a suitable example.
3. What is the difference between data and information?
4. What is an ICT system and what is its purpose?
5. Explain the terms data warehousing and data mining.
6. What is a POS system?
7. Give three benefits of using barcodes and scanners in a POS system.
8. Many schools have some form of ICT system for their pupils. List some of the areas where information can be extracted from the data stored in such a system.

Glossary

Application software	Software that is used to do productive work and accomplish specific tasks.
Computer	A multi-purpose electronic tool that can receive data (input), can process the data, produce results and output the results.
Data	Raw, unprocessed facts.
Freeware	Software which is distributed freely. You do not receive the source code. There is no limitation on the time or functionality included.
Hardware	Physical components and devices, the parts of a computer that you can physically touch.
ICT system	An ICT system is a combination of hardware, software, data, processes and *people* with the purpose to collect, manipulate, convey, store and retrieve data and information.
Information	The result of processing data and it should be useful and meaningful.
Input	The way of getting both data and instructions into the computer.
IPO model	Input, Processing, Output model.
IT and ICT	*Information Technology* (IT) is a term that refers to all the technology involved with the collecting, processing and storing of data/information. *Information and Communications Technology* (ICT) is a term that expands IT to include the transmission of data using communications technology. It includes technologies such as computers, cell phones, the Internet and other networks as well as broadcasting technologies (radio and television).
Open source software (OSS)	Software made available with a license that allows you to access and modify the source code, but not to sell the software
Output	A way of allowing the computer to show the results of the processing it has done
POS system (Point of Sale system)	Specialized software and hardware designed to be used at the till point. It manages stock and creates reports (such as a till slip) when stock is sold to customers.
Proprietary software	Software developed by a certain software house. Users have to pay a license fee to be allowed to use the software. No access to the source code is provided.
Shareware	Software which is distributed freely, but either it can only be used for a limited time, or limited capabilities are available. Also you do not receive the source code.
Software	The programs – sets of instructions – that that tells the device/computer what to do.
Source code	The programming code used to develop a specific application.
System software	Software that is used to control and maintain your computer.

MODULE 2.2

Basic Concepts of hardware

When you started learning about the basic concepts of computers, you also learnt a little about computer hardware. In this module we are going to expand on this to give you a better idea of how the different parts of a computer really work and what they can be used for.

We will divide our attention between that which is *inside the case* and *peripherals* (all the devices that connect to the computer case).

Computers have a **modular** design.

This means that computers are assembled from a variety of parts which makes maintenance and upgrading easier than if they were made from just one piece. There are separate parts inside the case and a large variety of peripherals outside the case that are used for input, output and storage.

Inside the case of a computer

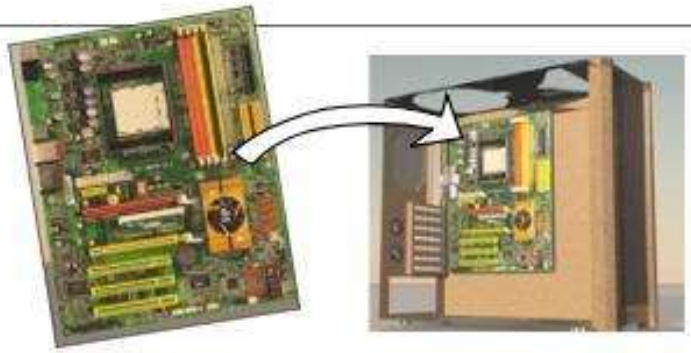
The case itself is just a metal enclosure. It is possible to build a computer inside any container. Metal is usually chosen because it is durable and provides good protection.

You could build a computer in a cardboard box if you wanted to. Some people build their own special cases (called mods) and come up with some pretty creative ideas. Search the Web for 'computer case mods' to see some of the weird and wonderful computer cases people have created!

Inside this case we find a number of components. In grade 10, we are going to focus on the CPU, RAM and storage units.

The **motherboard** is a large piece of circuitry with slots and connectors for different components and peripherals to connect to it. It allows all the parts of a computer to communicate with each other.

The motherboard fits inside the case so that expansion slots and ports line up with holes at the back of the case.



The CPU and cooling fan

The CPU (Central Processing Unit) is the part of the computer that does all the work (processing).

The CPU plugs into a ZIF (Zero Insertion Force) socket on the motherboard, which is designed not to damage the delicate pins on the CPU.

The CPU gets very hot when the computer is on. Overheating can damage a CPU, which is why all CPUs need a cooling fan.

The CPU and cooling fan on the motherboard.

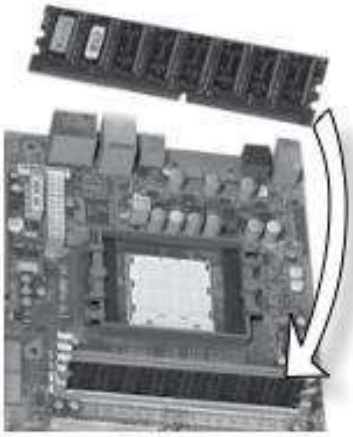
Personal computers use CPUs made by Intel or AMD, whilst smartphones and consumer tablets usually have CPUs made by ARM, Qualcomm, Apple and other manufacturers.

The speed of a processor is given in terms of Gigahertz (GHz). The speed rating is only one aspect that affects the overall speed and performance of a computer.

Everything else in a computer system is designed to get data and instructions to the CPU so that it can process them and produce information. This information can then be stored, communicated, or given as output in some format.

RAM

A DIMM is shown at the top of the motherboard whilst another DIMM is plugged into the slot at the bottom of the picture. Most motherboards do not have more than 4 DIMM slots.



RAM is often just called 'memory'. RAM stands for *Random Access Memory*.

Before any program can run or any data can be processed, it has to be loaded into RAM first.

There are a few things you need to know about RAM:

- It is *volatile* (it loses its contents when it has no power).
- It is where the CPU keeps all the data/instructions it is working on (a CPU can not work without RAM).
- Data transfer between RAM and CPU is fast because it is an electronic process.
- The more RAM you have in your computer, the better it will perform.
- RAM is supplied in DIMMs (Dual Inline Memory Modules) that fit into DIMM slots on the motherboard.
- RAM is expensive (it costs more per GB than storage).

Storage devices inside the Computer

A computer has to have some form of storage to be able to keep the programs and data files it needs when it is switched off. Remember, RAM is volatile and relatively small. When the computer is switched off, RAM loses its contents. You need some more permanent way to keep your programs and data so that you don't lose them! That's why computers have to include storage.

Storage inside the computer normally takes the form of *magnetic* and *optical* storage *(Note:* storage is not memory).

Read-write heads move in and out whilst the disks spin beneath them. HDD technology is being improved by increasing the density of data on the surface of the disk. This means that more data can fit in the same space (and at roughly the same cost).

Magnetic storage: Hard Disk Drives (HDD)

Hard drives are solid disks coated with a magnetic substance. The drive includes a read-write head that can either write magnetic patterns that represent 0s or 1s on the disk, or read the pattern from the disk.

Two sides of the same hard drive. The bottom contains the electronics necessary to control the HDD.

Some facts about HDDs:

- They are mechanical and therefore slow (much slower than RAM).
- They have large capacities at much lower cost per GB than RAM.
- They are delicate and can be damaged if moved whilst they are working (the head can make contact with the surface and damage the magnetic material).

Optical storage: CD, DVD and Blu-ray

Optical drives use removable discs (CD / DVD / Blu-ray) that hold data by reflecting laser light in different ways to represent 0s or 1s. Some things to know about optical disc drives:

- They are slower than hard disk drives.
- They are limited in storage capacity – around 700 MB for a CD, 4.7 GB – 9 GB for a DVD, and up to 50 GB for a Blu-ray disc.
- You can buy writeable or re-writeable blank discs and write data onto it yourself.
- You can buy them pre-made with programs/data already on them – for example when you buy software in a shop. (These days many people download software from Internet and pay online.)
- They are being replaced by flash drives and portable hard drives (more about them later) as a way of carrying data around.

Most computers include at least a DVD writer. DVD writers can read and write CDs as well. Blu-ray is still very scarce in computers, and most computers are only likely to have a Blu-ray reader (i.e. they cannot write to blank Blu-ray discs).

With optical drives, a tray slides out for the user to insert and remove a disc to be read/written.

CDs, DVDs and Blu-ray discs look identical at first glance – you need to check on the inner ring of the disc to see which it is.

Solid State Drives

Solid State Drives (SSDs) use a special type of electronic memory that can retain its data when the power is off (they use the same technology that is used in your flash drive). They are slower than RAM, but, because they are electronic rather than mechanical, they are a lot faster than mechanical storage like magnetic and optical drives.

They are much more expensive and have a much smaller capacity than traditional magnetic hard drives but they are much, much faster. You find them in some high-end laptops and similar technology is used to provide the storage for devices like Smartphones and consumer tablets. They are slowly increasing in popularity.

Connecting to the Computer

The case with all its parts inside it is pretty useless unless it has some way of communicating with the world. That's where *peripherals* come into the picture. Peripheral devices give the computer the equipment it needs to get input and provide output. We will talk about peripherals in a short while, but before we get there, the peripherals themselves need some physical way to connect to the computer.

USB

We are quite lucky in that today most peripherals can be connected to a computer by using a single connector (usually called a port) on the computer. That port is the USB port. This is why most computers have 3 or 4 (or even more) USB ports.

Which peripherals connect via USB?

The list is nearly endless – keyboards, mouses, printers, scanners, drawing tablets (like the Wacom Bamboo), flash drives, external hard drives, external CD/DVD drives, cellphones, cameras, microphones, headphones, speakers, GPS devices, cup warmers, LED lights – in fact, if you can think of it, it probably exists somewhere and uses USB to connect to your computer!

Tip: If you have a laptop, then you should really get a hub and plug everything into the hub. Then you only need to plug one USB cable into the laptop and, presto! – all your devices are connected.

USB hub

Because of the large number of devices that use USB ports, you might find that you run out of free ports before you have connected up all your devices. The USB hub connects to a USB port and makes 4 or more new ports available to plug devices into.

Firewire

The earlier versions of USB are not fast enough to transfer the huge amounts of data involved in getting video from a device such as a video camera onto your computer. Therefore, historically, the connection of choice for video has been Firewire. Firewire has always been more popular on Apple Mac computers. It is, however, falling by the wayside and has been recently replaced by a new technology.

Thunderbolt / Lightpeak

2011 saw a new connection port technology reach the consumer market to compete with USB. Intel developed this technology under the name of Lightpeak, and it will be available for all types of PCs and Macs. It was first released as 'Thunderbolt' in 2011, and is available on all new Apple computers. Thunderbolt provides users with the fastest external connection currently available.

Other Connections

Computers also need to have

- a way to connect to a cabled network
- sockets for normal headphones/speakers and microphones
- a port to connect to a monitor.

These connections are simple to see and recognize:

- The wired network uses an RJ 45 connector (if you want to use wireless, the connection is not a visible port).
- Headphones and microphones use standard mini-jacks.
- The monitor generally uses a standard VGA connector.

All of these connectors are shown in Appendix A.

Connections and connecting

The development of fast connection ports like USB, Firewire and Thunderbolt mean that you almost never have to open the case to add powerful peripherals to your computer. The only exception is if you want to upgrade your RAM or CPU. Adding extra storage using external drives is simple, fast and easy.

1 Activity

1. What is meant when we say that a computer has a modular design?
2. Explain the advantage of a modular design.
3. Name two functions of the motherboard.
4. What is the role of RAM in a computer?
5. Explain the difference between optical and magnetical storage.

6. An advertisement for a new computer gives the following as one of its specifications: Intel i7 2.1 GHz Quad-core.
   a) Which component is being referred to?
   b) Name another company besides Intel which makes this type of component.
7. Why is there a fan inside the computer case even if the case does not feel hot?
8. List six devices that can be connected to the USB port.
9. How can you add more USB ports to your computer?
10. What is the Firewire port mostly used for?

Peripherals

Peripherals are all the devices outside the case or system unit that connect to the computer via a port such as the USB port. Peripherals can be examined using the same structure as the model of the computer (i.e. input, output, storage and communication).

Input devices

There are many input devices used by computers. We can't possibly look at all of them, so instead we are going to look at the most common ones – and see how they are used to digitise different types of data.

Keyboard

Your average keyboard is what is called a QWERTY keyboard, after the order of keys in the first row of letters. You press keys to enter characters. Keys can be pressed simultaneously, and additional special keys like 'Alt' and 'Ctrl' allow for extra functionality.

Touch screen devices can display a 'keyboard' on the screen for entering text. You can also get 'projected' keyboards that project 'keys' onto any surface using LED or laser and use sensors to 'see' what keys you press. Many of these are gimicky and don't work very well – and ultimately they are all 'keyboards' used for getting text into the computer.

Keyboards connect using a USB port, or wirelessly, using radio technology such as Bluetooth.

Pointing devices

A **mouse** is a device that you move around on a surface, and a cursor on the screen moves in sync with the device. You point at items on the screen and click, right-click, double-click, drag and scroll to interact with the items on the screen.

Touchpads (trackpads) do the same thing, but are usually found on laptops, and you move your finger on the surface to accomplish the same tasks. The surface responds to the electric conductivity of the human finger and moves the pointer on the screen just the same as when you move a mouse. Touchpads use the same sort of technology as the touch screen used on smartphones and consumer tablets.

Microphones

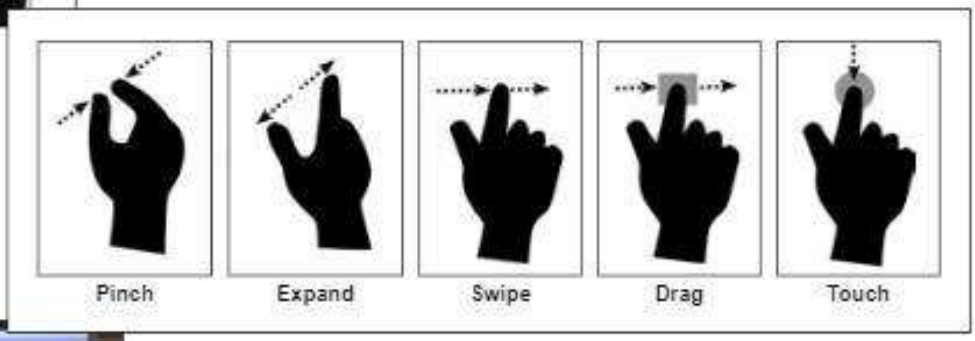
Microphones can be built in (most portable devices like laptops, smartphones and media tablets have microphones built in) or can be added by plugging in a microphone to a microphone jack / USB slot.

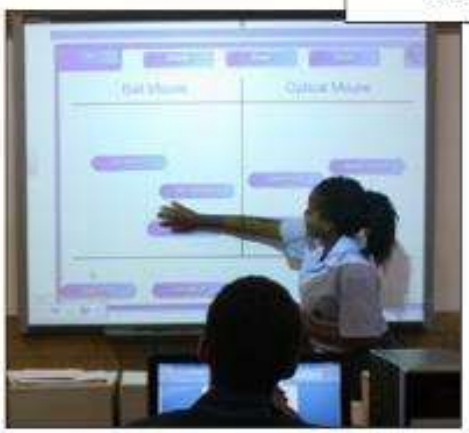
Microphones enable your computer to 'hear' – i.e. to be sensitive and respond to sound – or simply to record it.

Touch screens

Touch screens are usually found on consumer tablets and smartphones, but are available on some computers as well.

Instead of a cursor and a mouse, you directly touch the elements on the screen you want to interact with. This enables you to control the computer with gestures. For example, you can pinch with two fingers to zoom out, swipe to scroll, etc. Some gestures are shown below:





Touch screens are built into the screen of the device. Capacitive touch screens respond to fingers (or objects that conduct electricity), whilst resistive touch screens are best used with a stylus. Note that these types of 'touch screens' that use a stylus are not the same as 'pen input' devices – the stylus is just a substitute for a finger. Real 'pen input' devices (Pen Tablets) are discussed below.

Interactive whiteboards are pointing devices that work best when a projector is used to display the screen on the whiteboard – then the whole setup works like a giant touch screen! These are frequently used in educational settings.

Game controller

Playing games needs different types of control options than can be provided by just a keyboard and mouse (or, if you are using a console, you don't have a keyboard or mouse), so you have specialized controllers like joysticks or specialised game controllers such as those used for the Xbox or PlayStations.

You even have motion-sensitive controllers like those used in the Wii, and cameras that watch you and that then treat your body as a 'controller', like the Microsoft Kinect for the Xbox.

Camera and video camera

Often you want to be able to get your photos and videos onto the computer to be able to edit them, make calendars, posters or collages, or just to share them with friends.

Luckily most cameras and many video cameras are digital (they capture and store pictures and video as computer readable files) allowing you to easily transfer their files to the computer by

- connecting the camera to the computer with a cable
- removing the storage card in the camera and placing it in a card reader connected to the computer.

Scanners

Before the arrival of the digital camera, the only way to get a picture onto the computer was to use a scanner. You place what you want scanned onto a flat glass surface, and a light and sensor moves from one end of the surface to the other, capturing the image of your item. The computer then converts this input to a picture.

Special software (OCR or Optical Character Recognition software) can take a picture of a document and convert it into editable text.

Besides optical scanners that use light to read images, you also get other types of scanners or readers.

- You also get **3D scanners** that use lasers or multiple cameras to scan all around an object and create a 3D model of it in the computer.
- **Barcode scanners** are commonly used to quickly identify items by scanning barcode images and converting them into alphanumeric ID codes (you see them in supermarkets).
- **Magnetic stripe readers** read the information stored on a magnetic stripe, such as the stripe on a bank card or credit card.
- Finally you have **biometric scanners** (usually optical) that are specifically designed for security, and read and identify unique identifying charactersitics of people, like fingerprints or the pattern of an iris. These are more secure security devices than using passwords, keycards or PIN numbers, but they're not infallible either.

Fingerprint scanner on laptop

Pen Tablet

The pen tablet (or the Tablet PC) allows you to draw or paint in precise detail on 'digital paper' (a tablet or screen) as if you were drawing or painting on physical paper. This technology uses a pen or stylus that is far more precise and accurate than using a finger.

The tablet generates an electric field. The pen transmits its precise position and the pressure you are applying. The device is pressure sensitive (press harder and you get a thicker line) and is the ideal tool for anyone who wants to create digital art, as it allows for much more natural and realistic drawing than what can be created with a mouse.

Other devices

Besides the devices discussed you have the endless varieties of buttons, scroll-wheels, toggle switches, rocker switches, etc. that you find on everyday electronics such as GPS, MP3 players, cellphones, toys, gadgets and household appliances.

Go beyond the obvious!

What you really need to understand about input devices, is that they are used creatively and in unexpected ways to do far more than just what you think they are obviously used for. The section below is important because it helps you to get an idea of only some of the ways that input devices have been used innovatively and creatively to improve the way that computers work.

Types of input data

Before we look at types of data, you must be aware that input is not just data.

- Input is used to tell the computer what to do – i.e. input is used to give the computer **instructions**.
- Input gives a computer program the **data** that it needs to process.

Giving the computer instructions

Most of the controling (giving instructions) of computers on modern devices is done using *pointing devices* (i.e. mouse, touch screen, etc).

We can also use *voice* (to a limited extent) to give basic instructions (this works better on devices like smartphones than desktop or laptop computers). It's really useful to be able to tell your smartphone to 'Call John Edwards' rather than to have to scroll through the address book or type in the number.

As an IT student, you need to be able to think about what is meant by 'giving instructions', as compared to 'inputting data'. You also need to be able to identify the type of data (text, sound, image, etc.) that needs to be input and the best ways to get that data into the computer.

Now let's look at the really interesting stuff – the different types of data that can be input into a computer.

Text

The keyboard is the main device used for inputting text. But is a keyboard always the best (or only) solution? Often large amounts of text data is captured using anything but a keyboard.

Images from cameras and scanners, as well as voice can be processed to recognize text and convert it into editable format. Here are some amazing examples of the power of this concept:

- There's low cost software for smartphones that allows you to take a picture of a sign in Spanish with the phone's camera. It then converts the sign to text, translates it and gives you the English version.
- Automated batch scanners can take huge piles of documents, scan them and convert them to text at high speed.
- Voice recognition software allows you to dictate to the computer, and converts your spoken words into editable text.
- Barcode scanners turn barcodes into text codes. You can then look up the code in a database to get extra information on the scanned item. You even get smartphone software that uses the phone camera to take a photo of the barcode, identifies the item, and then lists comparative prices for it from stores on the Internet (so you can easily find out if you can buy it cheaper!).

Images

Did you know?

The increase in processing power means that computer can actually perform complicated analysis and interpretation of images instead of using numerical data, and therefore that pictures are becoming a more often-used type of input. Consider the following examples:

- The iPhoto application on Apple computers includes facial recognition features so that it scan through your photo library and quickly show you all photos that include a specific person, simply by recognizing that person's face.
- Kinect uses two cameras to track the way your body moves to control games played on the Xbox.
- Additionally, camera sensors are increasingly being used in industrial and military applications processing image data faster and more accurately than humans can. This is incredibly important, as the data captured by camera and other sensors make amazing advances in technology such as a self-driving car possible! (Do a search – they really have made a car that can drive itself in city traffic without causing accidents.)

Many computers and computing devices have built-in cameras. These can be used to capture video/photos for tasks such as video conferencing. We have also seen in the Text section how these images can be used as notes, or for barcodes, etc.

Scanners also capture images for use on the computer, whilst pen tablets allow you to draw images directly in image editing software instead of having to draw on paper and then scan into the computer.

Images/video can also be transferred from memory cards used in digital cameras, video cameras or even smartphones – or even by directly connecting these devices to the computer itself.

Sound

Sound is most often simply recorded, but computers can use the input from a microphone to allow you to dictate text or give it instructions. MiDi (Musical Instrument Digital Interface) instruments can be connected to the computer to help with composing or writing music. Software such as *Sibelius* will allow you to play music on a keyboard connected to the computer and write out the sheet music!

Sound can also be analyzed and manipulated. For example:

- There is software that will allow you to leave your laptop, smartphone or tablet in a room, and only record when the sound it detects is above a certain level of loudness (meaning it will only record when someone is actually speaking), saving battery life and preventing recordings with long periods of silence.
- Note taking software can link recorded sound to the notes you make, so that, for example, in class, you record the lesson whilst making notes. Later, if you click on or touch a specific note, the recording jumps to the section when you made the note (no having to rewind / fast-forward to find a place).
- Software like Shazam and Midomi Soundhound allow you to let your smartphone/tablet 'listen' to a few seconds of a song, and it can then tell you the title, artist, lyrics and even link to the song on YouTube and on the iTunes/Amazon music stores.

Data from sensors

Many devices include additional sensors such as an accelerometer (*g-sensor*). Accelerometers are the sensors in smartphones that allow you to control games and do things like 'undo' or 'skip this song' simply by shaking your device. There are also light sensors, gyroscopes, temperature sensors, etc that allow devices to react to their environment in ways that were not possible before. Examples are

- laptop screens change brightness according to how dark or light the room is
- some equipment can shut themselves down when their temperature becomes too high.

2 Activity • Input

1. When purchasing a computer system, you also need to look at the peripheral devices that are included in the price.
   Explain what we mean by the term 'peripheral' devices by referring to two common examples.

2. What type of port do most peripherals connect to?

3. A friend saw an advert for a slide-out 'full QWERTY' keyboard for his iPhone.
   a) What do we mean by the term 'full QWERTY' keyboard?
   b) Why would a full keyboard be of benefit when sending an SMS?

4. A friend says that the *Ctrl* key on his laptop's keyboard is not working, as it 'does nothing' when he presses it.
   Explain to your friend why his assumption that the key is not working is probably wrong, by referring to a typical example of how this key is typically used.

5. Which type of computer is more likely to have a *Bluetooth keyboard* – a laptop or a PC? Briefly motivate your answer by explaining what a 'Bluetooth keyboard' is.

6. A friend came across an article describing virtual or 'projected' keyboards.
   What are virtual or 'projected' keyboards?

7. Touchpads are usually found on laptops.
   Explain why this is the case by referring to the function of a touchpad.

8. Touch screens provide similar functions to a traditional mouse in that you can select and move objects.

   What other type of functions do touch screens provide, that are not supported by traditional mouse operations?

9. Explain what interactive whiteboards are and why they are so frequently used in teaching environments.

10. What is a motion sensitive controller, and where would they typically be used?

11. The Science Club at school have recorded a number of experiments on a digital video camera to place on the school's website.

    Give two common ways to transfer these videos to a PC so they can be edited.

12. Your teacher printed a copy of an exam paper that was stored on his flash drive and then accidentally deleted the electronic version. He noticed some errors on the hard copy (printed) version and needs to make some changes, but he doesn't want to re-type the 20-page document!

    Describe the process he would have to follow to be able to reconstruct an electronic copy of the document that can be edited.

13. Explain what a 3D scanner is by referring to a suitable example.

14. A company suggested that the school use biometric scanners to control access to your school.
    a) Explain what a biometric scanner is by referring to a suitable example.
    b) What advantage would the use of biometric scanners provide in this case?
    c) What practical drawback would there be to using biometric scanners in this situation (besides any cost factors)?

15. Why is a Tablet PC an ideal tool for anyone who wants to create digital art?

16. Is data the only form of input in a computer system? Briefly motivate your answer.

17. Give three of the advantages of using barcodes for a supermarket.

18. Name three devices that can be used to capture images or video for input into the computer.

Output devices

Output devices are as important as input, but there is less variety of them because of the nature of human senses. We need to be able to see and hear the output a computer produces. (Computers can also give electronic commands to machinery and control them as a form of output, but that is seldom seen in general purpose computing.)

The devices that produce visual information are monitors and printers. Sound is produced by speakers and headphones.

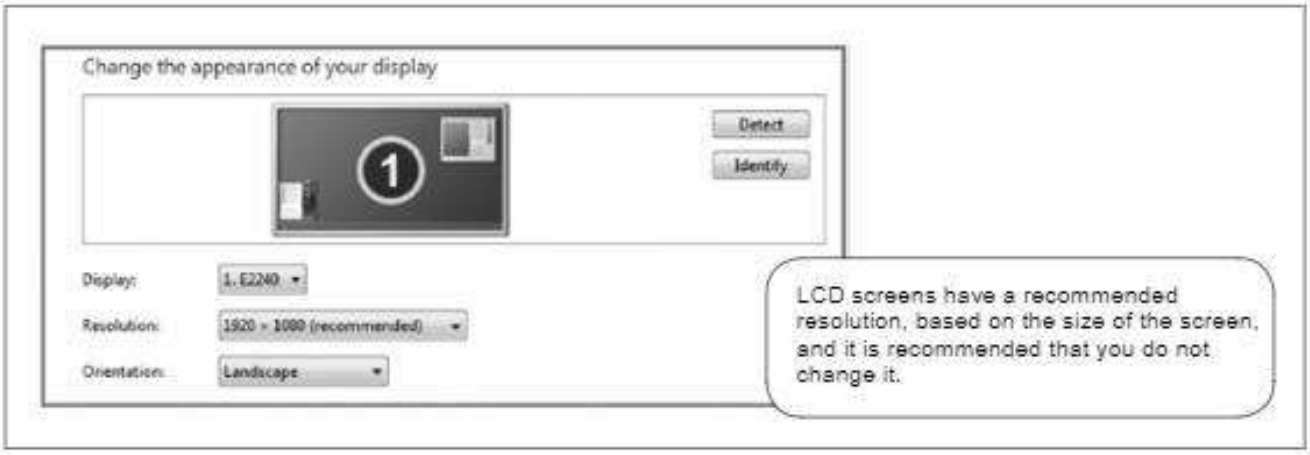
Monitor

The monitor or VDU (Visual Display Unit) is the device most commonly used to obtain output from the computer. Most VDUs today use LCD (Liquid Crystal Display) technology and many are using LED (Light Emitting Diodes) to provide the light that makes these monitors work. The advantages of this type of technology are that it is smaller, lighter, more energy efficient and cheaper than the older CRT (Cathode Ray Tube) technology it has replaced.

Monitors are typically classified according to their size, resolution, response time, the connection they use, the contrast ratio, and the technology used.

These are specifications you will see in a typical advert, and they are briefly discussed below:

- Monitors come in a variety of **sizes**. A common size is a 19' monitor, but larger sizes are also available. (The size of a monitor is measured diagonally in inches from one corner to the other.) Wide screens (which are much wider than they are tall – the same as high-definition TV screens) have become standard. The wide screen format is useful, as you can easily view documents side-by-side.
- Apart from the size, the **resolution** is also important. The resolution of the monitor refers to the number of pixels (small dots) used to form the screen image. The resolution is measured horizontally and vertically in pixels. A monitor with a resolution of 1920 X 1080 will display 1920 pixels across the screen, and 1080 pixels vertically ('down') the screen.



Note: Modern desktop monitors are called LCD (because of the technology they use) or flat-panel monitors. You may also see references to terms such as SVGA, XGA, WXGA, WUXGA, UXGA and WQXGA. These all refer to the resolution of the screen.

- For gamers the **response time** of a monitor is also very important. This is measured in milliseconds, and is the time it takes to change the value/colour of a pixel. A low response time is needed for high quality graphics and video in particular so the images do not blur or lag (ghost).
- There are three common ports or **connectors** for screens and monitors. VGA cables and ports are the most common, but also the oldest way of connecting a PC and a monitor. Digital Video Interface (DVI) is best suited for high-definition video, and HDMI is best suited for connecting a PC to a high-definition TV. Modern TV sets and computers have HDMI ports.
- The **contrast ratio** is the measurement or of the ratio between the darkest black and the brightest white the monitor can produce, e.g. 1000:1. The larger the ratio, the better.

The size and quality of the image displayed is only partially determined by the screen used. The image has to be generated by the video card inside the computer. High-end graphics requires a high-quality video card.

Visual information

Current computers have had an enormous impact on our creativity, largely because they are interactive devices. The VDU is the single device that is most responsible for this interactivity.

VDUs don't only allow for interactive control of the computer and provide you with immediate feedback – their development and advanced graphics means that you can see and interpret data in ways not possible before. Creating and viewing graphs and charts and diagrams, presentations, movies and slideshows has never been easier. All of this makes it possible to share and communicate and analyze data / information faster and more effectively than ever before in human history.

Data projectors

When speaking to large groups of people – when giving a presentation to an audience, for example – a monitor is simply not big enough or good enough. What we need is a 'data projector'.

A data projector connects to your computer and projects what is on your monitor on a screen or wall so that everyone can see a larger version of what would be on your monitor! The technology inside different projectors (i.e. how they create the image) can be different, and this results in them being called by different names, including LCD Projector, DLP, RGB Projector, LED projector and even a Pico Projector (a small low-power projector the size of a cellphone that only works in a dark room).

Printers

Printers are used to produce a hard copy. There are three main types of printers available:

Hard copy refers to information printed on paper, whereas *soft copy* refers to the electronic copy on the screen.

- **Laser printers** work by melting toner onto the paper. They are good quality printers – great for general purpose printing, faster than inkjets and relatively cheap to run.
- **Inkjet printers** spray dots of ink onto paper. They are slower than laser printers and more expensive to run, but they are the best option if you are trying to print photo quality images. Large inkjets are used to produce posters and banners, etc.
- **Dot-matrix printers** use very old technology that works by striking pins onto paper through inked ribbons. They are still used for printing invoices, bills, etc. and where carbon copies are needed.

Multi-function devices, which incorporate scanning, printing and e-mail features, are often used in the business environment, as they occupy less space and can be cheaper than purchasing separate equipment for each function.

Printers usually connect to the computer using USB, but some can be connected directly to a network (meaning computers on the network can print to them).

The environmental impact of printers is discussed in Module 5.2.

Speakers and headphones

Sound is produced using speakers or headphones. The quality of the sound that a computer produces depends on two things:

- The sound card
- The speakers or headphones used

More expensive, professional equipment will produce sound of a better quality than the sound card built into the motherboard and a cheap pair of USB-powered speakers.

Computers can also:

- be connected to electronic instruments to play sheet music on them.
- connect to professional mixing boards to control the sound and mixing of sound in recording studios and concerts.

Speakers and headphones connect using the headphone jack, whilst MiDi instruments, headphones and speakers can connect using USB.

Sound output can be used in many more creative ways than simply playing back sound for music or movie files. Some examples of how sound can be used includes:

- to provide the user with feedback for operations – e.g. notification sounds that tell you when a task starts / completes or when an error occurs.
- to enhance the user interface experience (e.g. sounds when you click on an object).
- to provide additional functionality for visually-impaired users – e.g. a text-to-speech synthesizer can 'read' the menu or text.

3 Activity • Output

1. Which two types of output can one get from a computer system?
2. Consider the specifications given in the following advert for a monitor and answer the questions that follow:

> 21.5" Wide LED Backlight Display
> 1920 x 1080
> 8 000 000:1 Contrast Ratio
> 300 Nits
> 2 ms response time
> VGA, DVI-D & HDMI inputs
> 3 Year Warranty

   a) Cleary explain what the '21.5"' specification refers to.
   b) The monitor is listed as a wide display. Explain what this means and give two benefits of a wide screen for the user.
   c) Cleary explain what the '1920 x 1080' specification refers to and what significance this has for the user.
   d) Explain what the '2 ms response time'-specification refers to and whether this would be important for a secretary in an office.
   e) What does the 'contrast ratio'-specification refer to?
   f) The monitor makes uses of LCD technology.
      Give two advantages this type of technology provides over older CRT technology.
   g) The monitor provides three types of connection: VGA, DVI and HDMI.
      What are the DVI and HDMI ports typically used for?
3. A friend purchased a high-end monitor for his older computer in order to play high-end graphics games. He was disappointed to note that he could not set his computer to match the maximum resolution of the new monitor.
   Explain why this might occur and how to solve the problem.
4. The principal would like the school to show the Springbok rugby game in the hall on Saturday but the school's television sets are too small and all the big-screens have been hired out.
   Explain how a data projector could be used to solve this problem.
5. Name the three main types of printer available today.
6. What type of printer would be best suited to printing photo quality images?
7. What are the two main benefits laser printers have over ink-jet printers?
8. Dot-matrix printers make of relatively old technology, similar to typewriters.
   Give a typical use for these types of printers.
9. A sales representative suggested to your father that he buy a multi-function printer
   a) List two other functions, besides printing, that a multi-function printer provides.
   b) Give two benefits of purchasing and using a multi-function printer.
10. Which two hardware items determine the quality of sound that a computer produces?

Storage

When talking about storage as a peripheral you are looking at a very important aspect concerning the way we expect to be able to use computers. We want to be able to take our data from place to place quickly, easily, securely (i.e. the data is not easily damaged) and using as little physical space as possible.

Of the storage devices installed inside the case, only the optical disc drives (CD / DVD / Blu-ray) are designed with portability in mind. You can write data to a disc, remove it and go and insert it into any other computer (which has a suitable drive).

Optical storage is comparatively slow, has limited capacity and is and expensive per GB compared to magnetic storage or even Solid State Disks (SSDs). Discs are easily damaged by being scratched or exposed to heat. They are also comparatively large. As a result, the current trend is for portable storage not to use optical discs but rather to move to

- flash drives
- portable hard drives.

Flash drives

Flash drives are storage devices that use electronic memory to store their data. They are

- electronic rather than mechanical
- much faster than optical or magnetic storage
- comparatively more expensive per GB than magnetic storage
- less likely to be damaged compared to hard drives and optical discs.

Many people now work on computers owned and configured by their employers. Often the employer has strict restrictions on the software that can be installed and used on their computers. Technology such as the now obsolete U3 smart drive and the 'Windows to go' initiative are designed to allow you to install and run software from a specially configured flash drive. The software is never installed on the computer but on the flash drive, along with all its data and settings. When you plug the flash drive into any computer, the software is ready to use even if it is not installed on that computer's hard drive. This solution is also used by people who move between different computers frequently – allowing them to make sure they can use their favourite software wherever they are.

The need for this technology has been greatly reduced by the fact that people are using more laptops than desktops and so have their personalised computer with them all the time.

Flash disks range in size from 1 GB upwards (and are getting bigger in storage capacity, smaller in size and cheaper all the time). Because of their small physical size, high capacity and low cost, they are very popular as portable data storage peripheral. One of the potential downsides of portable storage is that they get stolen or lost.

Flash disks get their power from the computer, through the USB port they connect to. To use a flash disk, you simply plug it into the USB port and you will see it appear as another disk. It is normally listed as a 'Removable Disk' or 'Flash Disk'.

Memory cards

Digital cameras store their pictures on memory cards. There are many different types of memory cards (SD or Secure Digital, Memory Stick, Compact Flash – to name a few) and they all have different sizes and different types of connections. It is possible to get the pictures onto your computer by connecting the camera to the computer with a cable (usually USB), but this can waste your camera's battery power, and it also often means you have to install special software.

Flash memory provides the storage used in integrated devices such as cellphones, smartphones, MP3 players and GPS devices. It is also the technology used in the memory cards found in digital cameras.

A much better option is to buy a 'card reader'. This allows you to plug the card into the reader and then, when you connect the card reader to a USB port, the computer treats it like a flash drive, and you can simply copy the photos like you would copy any other file! (Some computers, especially some laptops, already have a card reader built in.)

Portable hard drives

Portable or external hard drives use the same magnetic media used with internal hard drives. They connect to a computer via a USB, Firewire or Thunderbolt connection. Portable hard drives have one advantage over flash drives – they have much greater capacity than flash drives. This makes them better suited for backups and carrying around large amounts of data like media collections.

You get two types of portable or external hard drives, namely:

- 2.5' drives
  These drives have a lower capacity than the bigger drives installed inside a desktop computer, but have the advantage that they need less power, and so can run on the power that is supplied via the cable.
- 3.5' drives
  These drives have much larger capacity than 2.5" drives but have the disadvantage that you have to carry around an extra power supply and need access to a power socket before you can use them.

Communication

The first device that was used for communicating between computers over a distance was a ***modem*** *(MO*dulator – *DEM*odulator). This was a device that converted a computer's digital data into analogue audio signals, so that they could be transferred over the telephone system – which was designed to carry audio signals. When the sound reached the target modem, it was converted back into digital data. This method was used because the telephone network and its infrastructure to send data over long distances was already in place.

These original types of modems are rare today, as they are comparatively slow. The problem is that everyone got used to calling a communication device a 'modem' – and so the newer technology, which does not even function like a modem did, is still called a 'modem'. In reality these are DTD's (Data Terminating Devices) that use different technologies to send and receive data over different media. Their function is to convert signals from your computer to the type of signals that can be transmitted over the telephone line and vice versa, so that communication can occur. Today data is transmitted digitally and does not need to be converted to and from analogue signals any more.

Examples of these include:

- 3G modems, which allow you to connect to the Internet using the cellular telephone network. They are ideal for people who move around a lot, but who still want to be able to access the Internet wherever they are. 3G Modems are built into some laptops and almost all Smartphones and consumer Tablets (such as the iPad).
- ADSL modems, which are used for high-speed, 24-hour-a-day connection to the Internet over a digital telephone network. These come at a fixed cost, and the user will still be able to have a normal telephone conversation on the same line at the same time.
- Routers, which are devices that make communication between networks over the Internet possible by helping to direct data to its correct destination.

4 Activity • Storage and Communication

1. Optical discs were once a very popular media for storing and transferring data from one computer to another.
   a) List two drives that make use of optical technology to store data on discs.
   b) Give four reasons why CDs and DVDs are no longer such popular storage media.
   c) Which two types of portable storage are increasingly being used instead of optical discs?
   d) How do flash drives differ from normal hard drives and optical drives in terms of the way they store data?

e) Give one advantage and one disadvantage of storing data electronically like flash drives do, compared to magnetic and optical storage devices.

2. Your dad decided to keep all his financial records on a portable hard drive in case his computer gets infected by a virus or the machine gets 'hacked'.
   Why it is not necessarily a good idea to do this?

3. Your sister is puzzled. She was told that flash drives store their data electronically but they have no 'power' connection. Explain to your sister how flash drives get their power.

4. Your sister has inserted a flash disk into her PC, but there are a lot of drives listed and she is not sure which one is the flash disk.
   Give her some advice as to how she can identify which of the drives is her flash drive.

5. Give two other examples of devices that make use of flash memory besides flash drives.

6. Name two different types of memory cards.

7. Your mother wants to transfer the family's holiday photos from her digital camera to her computer. One of the options is to transfer it via a cable connecting the camera to the computer.
   a) What port would this cable typically connect to?
   b) Give two possible disadvantages of using this approach to transfer the photos.
   c) What would be the best option to transfer these photos?
   Briefly explain how you would go about using the hardware or method you gave as an answer.

8. What did the original modems do?

9. Briefly describe what 3G and ADSL modems are used for.

Smartphones versus Computers

Comparison

More and more users are purchasing smartphones, and their sales are set to overtake desktop computers. But how do they differ, and can we totally dispense with our laptops or desktop PCs?

The answer (at the moment) is 'no'. Smartphones have the advantage of being completely mobile, and are perfect for tasks such as surfing the Net, e-mail and calendars. Yes, they do have scaled-down versions of operating systems and office suites, but they don't have the power of their bigger counterparts yet. In addition, you are hardly going to want to spend hours in front of a smartphone to type or edit long documents. You sacrifice screen size, processing power and storage for the sake of battery life and smaller size. The best solution is to have both a smartphone and a laptop or desktop, and use them as needed. The following table provides a comparison between a typical desktop PC and a smartphone.

	Desktop	Smartphone
Communication	Typically connects to an ADSL modem for Internet activity. Can also use a 3G modem to connect via a cellular network.	Will have built-in wireless connectivity such as Wi-Fi, Bluetooth and 3G
Input	Normally via a standard keyboard and mouse or a microphone for voice input	Usually has a small keyboard, touch screen and built-in microphone and camera.
Output	19" – 25' screen, speakers, headphones, can connect directly to printer.	Built in (3" - 5') screen, built-in speakers, headphones.
Processor	Intel / AMD multi-core processors.	Uses ARM, Qualcomm, Apple CPUs designed for lower power consumption and are slower than the equivalent desktop processors.
RAM	1 GB – 12+ GB	256 MB – 2 GB
Storage	Typically 250 GB +	8 GB – 64 GB

Synchronising your phone and your Computer

People often own more than one computing device. If you have a smartphone, it is likely that you also own a desktop computer or a laptop. The thing is that you use the devices at different times and make changes to the data on them at different times as well – and often you have the same data on both. Now one is out of date...

People find this problematic because they want to be able to access the most recent data on whichever device they are using. This creates a need to ***synchronise*** the data between devices (i.e. make sure that changes made on the one device are copied over to the other and vice-versa, and that each device has the latest copy of the data). This is important for data like address books, as it also creates backup of all your contacts' details should your phone go missing!

Most portable devices, such as cellphones and smartphones, have cables and software that allow them to synchronise their contents to a computer when they are connected to it with the cable (or even wirelessly, using Bluetooth or WiFi).

There a number of Internet services, such as DropBox, which allow you to store your files on their website, and you can synchronise your files by connecting to this website. This is sometimes referred to as 'Cloud Synching'.

5 Activity • Phone and Computer

1. What is the main advantage that a smartphone has over a desktop PC?
2. Smartphones also have operating systems and office suites. How do these differ, if at all, from the ones you would typically find on a desktop computer?
3. Give two reasons, from a hardware perspective, why you would not generally type a long document on a smartphone.
4. Smartphones are perfect for tasks such as surfing the Net and e-mail.
   a) What is the most common way for smartphones to connect the Internet?
   b) What type of connection might a desktop computer use to connect to the Internet, that is *not* typically used by a smartphone?
5. Name two hardware features or specifications that smartphone designers have to compromise on in order to extend battery life and reduce the size of the phone.
6. Your mother wrote down the following specifications of a smartphone from a salesperson at a local store:

   ARM processor

   17" screen

   2 GB RAM

   320 GB HDD

   Write down which of these specifications are obviously not correct, and suggest more appropriate specifications in each case.
7. People often have the same data on more than one computer or computer device, and the one 'version' gets out of date with the other.
   a) Why would someone have two versions of the same file in the first place?
   b) What is the name given to the process of ensuring that each device has the latest copy of the data?
   c) What do we mean by the term *'Cloud Synching'*?

Glossary

Biometric scanners	Biometric scanners (usually optical) are specifically designed for security, and read and identify unique identifying characteristics of people, like fingerprints or the pattern of an iris.
CPU (Central Processing Unit)	The part of the computer that does all the processing.
Data projector	A data projector connects to your computer and projects what is on your monitor on a screen or wall so that everyone can see a larger version of what would be on your monitor.
DIMM (Dual Inline Memory Modules)	Electronic circuit boards with RAM on.
Dot matrix printers	A very old technology using pins hitting ink ribbons to create images on paper.
Firewire	Devices can be connected to this port for fast data transfer. Mainly used for transferring video but is being used less and less.
Inkjet printer	Sprays drops of ink on paper.
Laser printer	Works by melting toner into the paper.
Magnetic storage	Stores data by means of magnetic fields, for example hard disks.
Modem	Device that converts the signal from a computer to a format that can be transmitted over a communication channel. 3G modem: To transfer data over a cellular phone network. ADSL modem: For high speed data transfer over digital telephone networks.
Modular design of computers	Means that computers consist of a variety of separate components for ease of upgrading and maintenance.
Motherboard	A large piece of circuitry with slots and connectors for different components and peripherals to connect to it.
Optical storage	Stores data by means of laser, for example CD, DVD, Blu-ray.
Peripherals	Devices connected to the computer.
RAM	RAM or Random Access Memory is where any program that is run is loaded and temporarily stored, and where files that are open are temporarily stored.
Router	Device that makes communication between networks over the Internet possible by helping to direct data to its correct destination.
SSD (Solid State Drive)	A type of disk that stores data electronically.
Synchronise	The process of getting two sets of the same data files on two different devices updated with the latest copy of the file(s).
Touch pad	Touch pads are rectangular pressure sensitive areas built into the laptop. They are input devices that can be used instead of a mouse.
USB (Universal Serial Bus)	Port that a variety of devices can be connected to.
USB hub	Connects to a USB port and expands it to 4 or more ports that devices can be connected to.

MODULE 2.3

Basic Concepts of system software

Introduction

You already know that software is divided into two categories, namely system software and application software. We are now going to focus our attention on gaining a better understanding of system software.

System software is software that is intended to control, support or operate the computer. The most important and well-known type of system software is the operating system (such as the Windows operating system).

Operating system

An operating system is system software which controls all activities that take place in a computer. It is what most people think of as system software. Most people also think of an operating system as a single program. This is not the case.

Role of the operating system

An operating system is a collection of programs that have the following functions:

- **Provides an interface**
  The interface allows us to interact with the computer and with programs on the computer. This is how we communicate with the computer. Modern software, including operating systems, all make use of a visual or graphical user interface (a GUI – pronounced goo-ey). We will expand on that a bit later.
- **Process and task management**
  Manage the CPU and how it runs programs – i.e. which programs and tasks can use the CPU and for how long.
- **Memory management**
  Manage the memory (RAM) to control how programs use it – i.e. make sure that programs don't try to use the same area of memory at the same time and so corrupt each other's data/ instructions.
- **Input/Output management**
  Manage input and output of the computer and how programs use the input and output devices – i.e. which programs receive input from input devices and which programs can use an output device at any time.
- **Disk management**
  Manage storage – both how the data is organised on the storage device and how or when programs access the storage device.

Examples and types of operating systems

Only multi-purpose devices need an operating system. Examples include PCs, laptops, smartphones and consumer tablets. Most dedicated devices, such as microwaves and washing machines, have a dedicated controlling program rather than an operating system.

Consider the following examples:

Operating systems for desktop computers (PCs) and portable computers

- Windows (created and owned by Microsoft, proprietary).
- Apple OS X (created and owned by Apple, proprietary).
- Linux (many flavours – Ubuntu is one of the best, Open Source Software or OSS).
- Google Chrome OS
- Unix (many flavours – both proprietary and OSS).

Operating systems for mobile devices

- iOS (for Apple devices like the iPhone, iPad and iPod touch, proprietary).
- Android (created by Google, OSS).
- QNX (for RIM products like the Blackberry Playbook, proprietary).
- Windows Mobile (a mobile operating system created by Microsoft, proprietary).

Category	Operating system
Stand-alone	Windows XP Windows Vista Windows 7 Mac OS X Linux
Network	Windows Server 2008 Windows Small Business Server Mac OS X LION Server Unix *Novell* NetWare
Embedded	Windows Embedded Compact Embedded Linux

Operating systems can be divided into three categories, namely stand-alone, network and embedded operating systems.

- Many desktop operating systems have a **server** version, which has extra features, giving it additional ability to control and manage shared resources, (e.g. files, printers) and users on a network. Server versions are often called **network operating systems**.
- **Embedded operating systems** differ from those installed and used on a general-purpose computer such as a PC, in that they are stored on internal memory chips. They are not loaded from disk as is the case of a computer with a hard drive. This makes them more difficult to upgrade.

Embedded operating systems range in complexity. You also find them on devices such as cellphones, barcode scanners, digital picture frames, e-readers, point-of-sale terminals and some gaming devices.

More about The Graphical User Interface

We said previously that one of the functions of an operating system is to provide an interface that we can use to interact with the hardware and software of the computer. By using a graphical user interface, and allowing people to work with the computer by 'pointing' instead of typing out command words, computers are more intuitive and natural to use.

What's important to note about a GUI:

- The user does not have to remember or type long instructions.
- It uses pictures (icons) to represent programs/instructions.
- It uses a mouse or some other pointing device to allow the user to interact with anything they can see on the screen.
- It makes using a computer interactive.
- The user can come to understand it and start to use it very quickly.

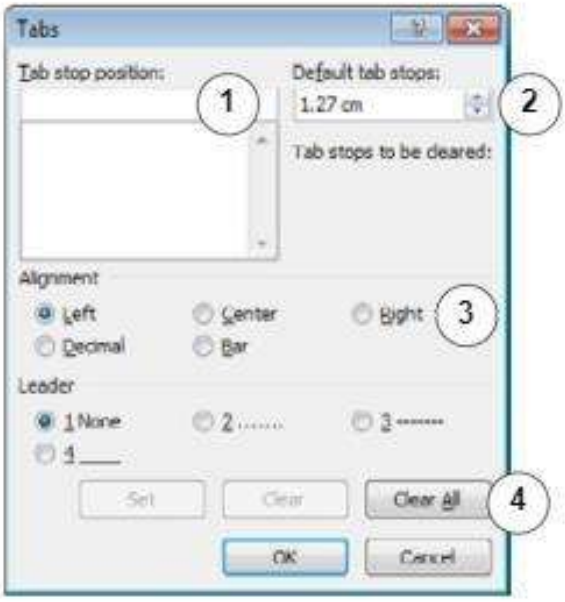
While different types of operating systems (including different versions) have different designs and interfaces, we interact with them in similar ways.

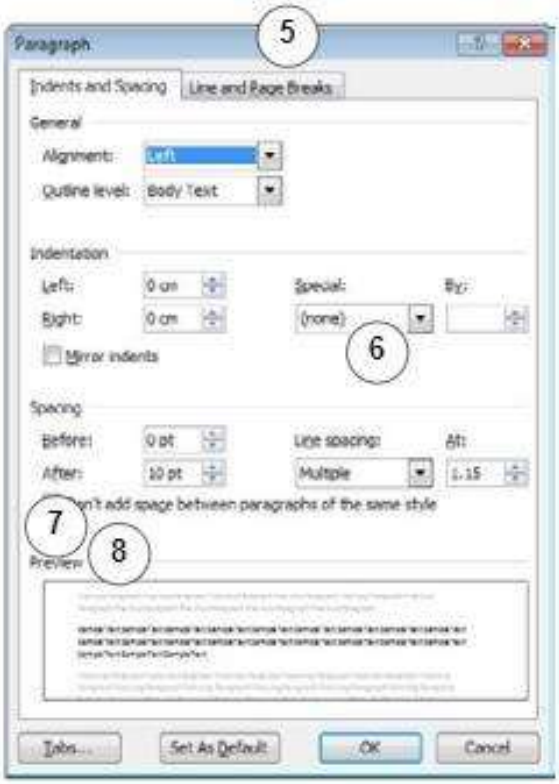
It is important to know both the terminology involved and some of the underlying concepts of how a GUI works. The reason for this is two-fold. Firstly, you won't be 'lost' when reading up on a feature or looking up something in a 'help' function and secondly you'll be using many of these techniques and controls in the very programs you are going to design yourself.

Windows software makes use of a number of commonly used components. The first one we are going to look at is the dialog box.

A dialog box is basically a small window that is used for communication or 'dialog' between a computer and a user. It usually provides options to a user for completing a small task. Dialog boxes can also be used to convey warnings or error messages.

Consider the following dialogs which are taken from *Word*. They provide options for formatting paragraphs and setting tabs. We have chosen these two to illustrate a number of other components.





	Control / GUI component	Function
1	**Textbox (Edit box)**	A control that allows the user space to enter information.
2	**Spinner**	A spinner can be used to change a value by clicking on an 'up' or 'down' arrow. The value of the spinner appears in a text box next to the spinner. The user can adjust the value using the spinner, or by editing the value directly in the text box.
3	**Radio button**	A radio button allows the user to choose only one from a given (small) set of options.
4	**Command button**	A command button performs a command (makes something happen) when you click on it. For example, clicking on the OK command button will mean that you are happy with your choices and that the dialog box can then close.
5	**Tab in dialog**	These controls often appear in dialog boxes so that different options can be shown on each tab or sheet. You can only view one tab at a time.
6	**Combo and List box**	These controls allow you to choose from a list of given options. They are identified by the ▼ appearing to the right of the control. Combo boxes differ from list boxes in that combo boxes allow the user to enter an option of their own, whereas list boxes force the user to only choose from the options listed.
7	**Check box**	Check box controls are similar to radio buttons in that they can be selected or 'deselected'. The difference is that, if there are few of them grouped together, more than one may be selected.
8	**Label**	Provide text to the user. Can be used to label another control, or provide instructions.
9	**Slider**	A slider represents a position along a continuum of values. The slider can be dragged to the desired position to indicate the 'value'. A typical application would be a volume control slider.

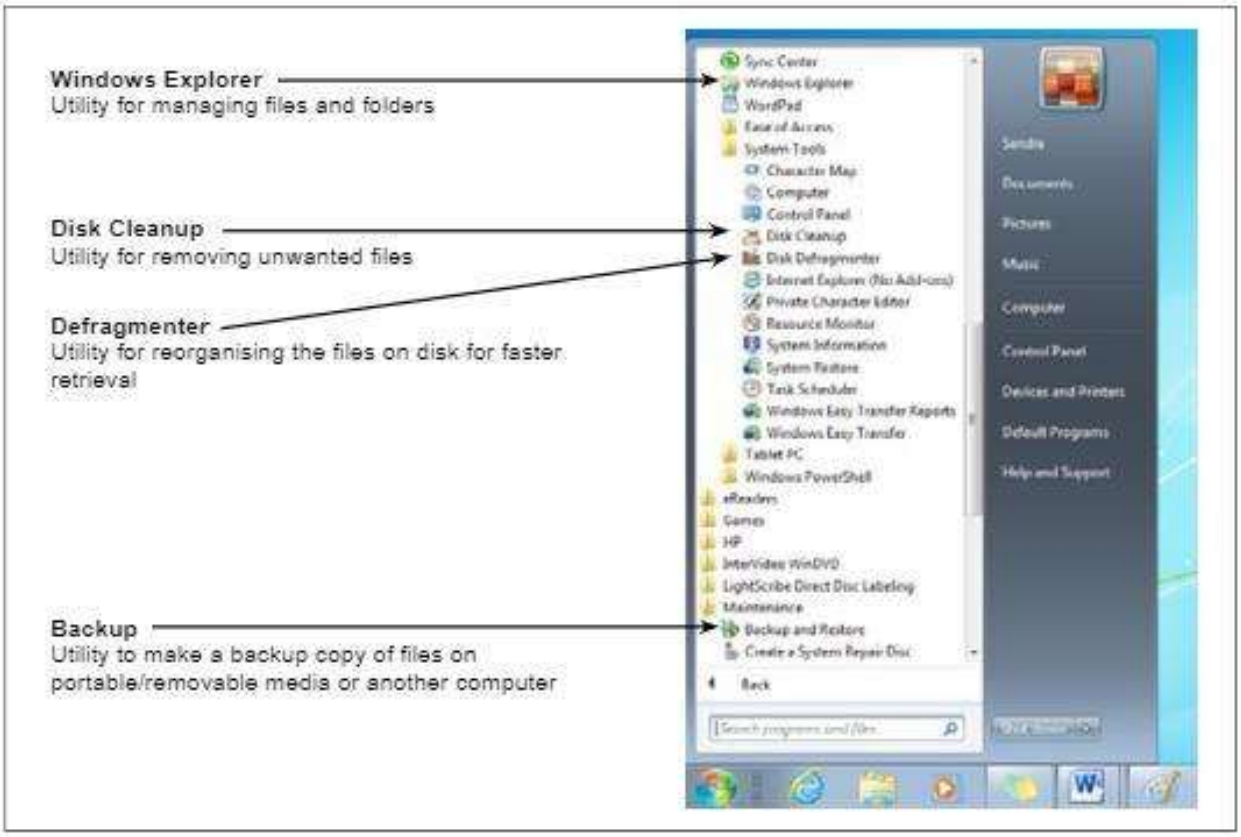
Utility programs

Apart from operating systems, system software also includes *utility software*. Utilities are programs that perform system maintenance and administration tasks. Several types of utilities are included with the operating system, but stand-alone packages can also be purchased separately.

Examples are

- Anti-virus software to prevent your computer from being affected by computer viruses.
- Compression software (such as WinZip) is used to make a compressed version of a file. This is aimed at reducing the size of a file so that it takes up less space to store. It is often used to reduce the size of e-mail attachments.
- Backup utilities: Backing up or making copies of our files on a regular basis is essential. Windows provides its own backup utility but a number of other backup utilities (some free) also exist.

A few utilities will be discussed in Module 2.4, Computer management.



Device drivers

Device drivers are essential pieces of software. They allow an operating system to communicate with and control a hardware device.

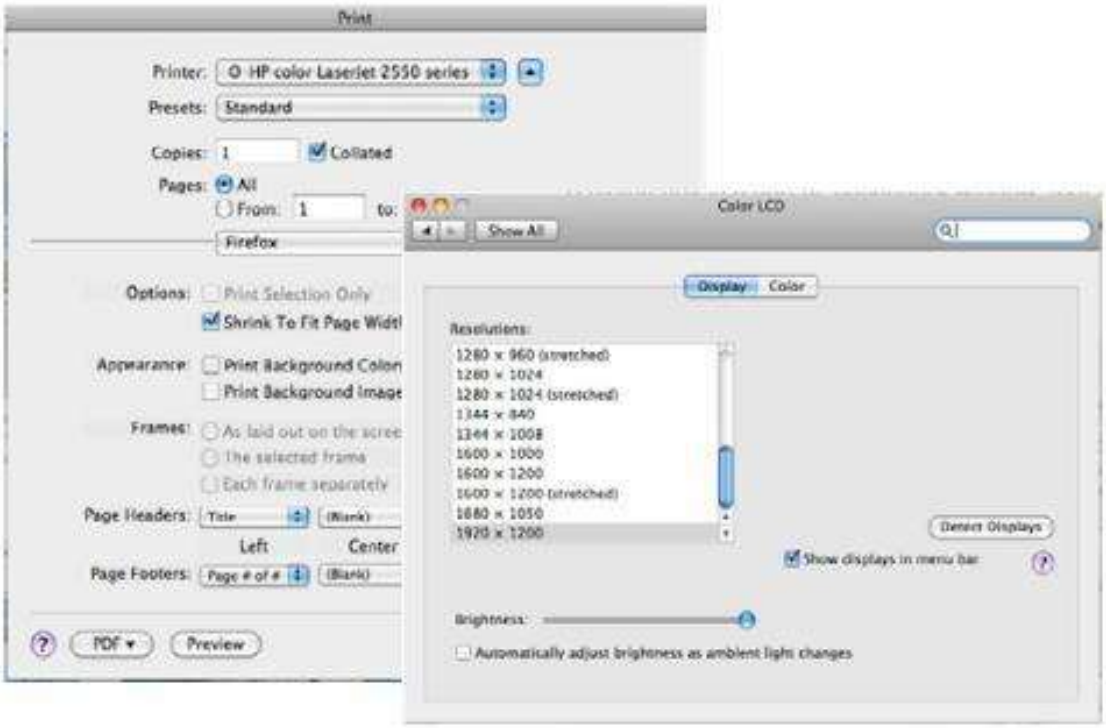
Operating systems usually come with a whole lot of drivers pre-installed, but new hardware comes out all the time. Hardware is usually supplied with a CD or DVD that contains the driver needed if the operating system does not already have a suitable driver. You install the driver, enabling your computer to use the hardware. Note that, in many cases, the new device will automatically be detected by the operating system, the driver installed, and the device configured for you automatically, so that it is ready to use.

If you lose the disc, you can usually download the driver from the manufacturer's website on the Internet. Drivers can be updated by manufacturers, and downloading and installing the latest driver for your hardware can actually improve your computer's performance and reliability.

1 Activity

1. Take a look at the same two dialogs in OS X and Linux and see if you can identify the same user interface element.

OS X



Linux (Ubuntu GNOME desktop)

2. Software can be classified as application or system software. Your father bought a new computer and says he bought application software like Office 2010. He did not, however, need to buy any system software as he is only going to use application software.
   a) What is the general function of system software?
   b) Explain to your father why he will be using system software even though he thinks he may not have bought any system software.

3. Give a concise (to the point) definition of an operating system.

4. One of the functions of an operating system is to provide the user interface. Modern operating systems and software in general make use of a GUI.
   a) Explain what we mean by the term 'user interface'.
   b) Explain what a GUI is – do not just give the meaning of the acronym.
   c) Give two other advantages of a GUI that you did not mention in your answer to the previous question.

5. The operating system performs a number of key functions besides providing the user interface. Two of these key functions are *memory* and *input/output* management.
   a) Briefly describe what each of these functions entails.
   b) List and briefly describe two other typical functions of an operating system, besides providing the user interface and providing *memory* and *input/output* management functions.

6. What do the following operating systems have in common?
   iOS, Android and QNX

7. Name the three broad categories that operating systems can be divided into.

8. Your mother wants to upgrade the operating system on her PC at home. She saw adverts for Windows 2008 Server and Windows 7. She thought that Windows 2008 server would be a better choice, as it is 'newer'.
   Comment critically on this reasoning by outlining the broad difference between these types of operating systems.

9. Your father phoned his company's help desk as he was battling to install some software. The person on the help desk asked him to wait until the first dialog box appeared on the screen.
   Briefly explain what a dialog box is.

10. Your sister who is in Matric has logged onto a university's website to try and register for her university studies next year.
    Explain how the following controls are used and give suitable examples of where they would be found on a web page in this specific scenario:
    a) List box
    b) Textbox (edit box)
    c) Command button

11. Explain the similarity and difference between a check box and radio button control.

12. Explain the similarity and difference between a combo and a list box.

13. Apart from operating systems, system software also includes utility software.
    a) What is the general purpose of utility programs?
    b) What would compression software (such as WinZip) be used for?
       Give an example to illustrate your answer.
    c) Give two common examples of utility software other than compression software.

14. Your father's business wants to upgrade the operating system on their machines from Windows XP to the latest version of Windows. A consultant suggested that they may have a problem finding updated drivers for some of their older printers.
    a) What is a device driver?
    b) Many users are unaware of drivers and have never installed them, despite adding new equipment like printers to their PCs. Briefly explain why this is the case.
    c) Where could your father's business try and obtain drivers for their older printers?
    d) Give a reason why one should try and update your drivers from time to time, even if you are not changing operating systems.

Glossary

Device drivers	Programs that allow operating systems to communicate with and control hardware devices.
Disk management	A function of an operating system whereby it manages storage – both how the data is organised on the storage device and how or when a program can access the storage device.
Embedded operating system	Embedded operating systems differ from those installed and used on a general-purpose computer such as a PC in that they are stored on internal memory chips. They are not loaded from disk as is the case of a computer with a hard drive.
GUI (pronounced goo-ey)	A graphical user interface makes use of visual controls to allow the user to interact with the computer and with programs on the computer
Input/Output management	A function of an operating system whereby it manages input and output of the computer and how programs use the input and output devices – i.e. which programs receive input from input devices and which programs can use an output device.
Memory management	A function of an operating system whereby it manages the memory (RAM) to control how programs use it – i.e. make sure that programs don't try to use the same area of memory at the same time and so corrupt each other's data/instructions.
Network operating system	An operating system which has extra features that give it additional ability to control and manage shared resources (e.g. files, printers) and users on a network.
Operating system	An operating system is system software which controls all activities that take place in a computer.
Process and task management	A function of an operating system whereby it manages the CPU and how it runs programs – i.e. which programs and tasks can use the CPU and for how long.
System software	System software is software that is intended to control, support or operate the computer.
Utilities	Programs that are part of system software and do maintenance and administrative tasks.

MODULE 2.4

Computer management

Introduction

Imagine walking into a house where no sweeping or vacuuming or dusting or dishwashing has been done for months – it would be unpleasant to live in such a place – and difficult (if not impossible) to get any work done in the mess!

Effective computer management or 'housekeeping' is a process which starts from the time you set up your computer. We can compare it with the concept of buying and maintaining a house.

Some of the tasks tend to be once-off tasks (like putting up a fence) while others need to be done on an ongoing basis (sort of like keeping your house painted and the burglars out!). Some tend to be very unique and personal – in this section we'll look at the most common issues surrounding maintenance and management.

Initial maintenance

New computers arrive with standard settings that will probably work for most users, but you need to be aware of them in case they do not suit you. This includes customising the system, setting its properties to suit your personal taste, and setting up security.

The best place is to start is to locate the Control Panel via the Start Button. You can use the Control Panel to change nearly every setting for Windows to suit your own needs and preferences.

Setting up accounts

User account

Each user account has a username and a password so that a user can only access his or her own account – for security purposes. In a networked environment, this will be set up by the network administrator. Users can then be assigned their own profiles which determine amongst other things

- what folders they have access to
- what changes they can make to the settings of the computer.

Administrator account

There is also a special administrator account which allows that account holder (user) to make any changes he or she wishes to the system, including to other accounts. This privilege, or right as it known, allows the administrator to perform tasks that other users may not be able to do, such as change security settings and install software and hardware.

It is not a good idea to use this type of account unless you have to. If you accidentally leave your computer logged in as an administrator, someone else may be able to intentionally or unintentionally cause havoc on the system! In addition, some viruses and other malware can spread a lot quicker via an administrator account as they have access and rights to the whole system, especially on a network! You would not want the whole world to have the key to your house!

Configuring the desktop

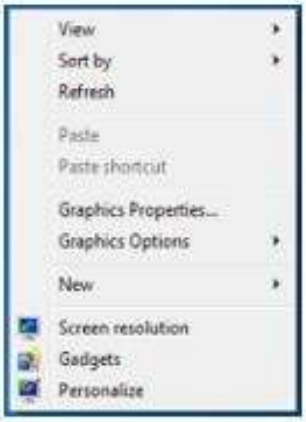
The desktop is the first basic 'screen' of the operating system. This will vary in appearance, because each operating system has a different appearance, and the desktop can also be customised or changed by the user.

The desktop can be compared to your desk where you can place your items, move or remove them as you please.

You can personalise the desktop settings in terms of the colours, appearance of the mouse pointer and even the theme you wish to use. Besides these personal preferrences, it is important to set the resolution of the monitor to a setting that suits you.

The Control Panel provides us with a number of options to change desktop settings as well as the screen resolution.

You can also right-click on the desktop to give you a pop-up menu which provides some options for customising the desktop.



Security settings

You need to make sure that all your basic security settings are in place and working. Think of it as making sure you have a basic burglar alarm in your house and that it is switched on! As a minimum, you should have a firewall and anti-virus software in place.

Firewalls

Whenever you connect to the Internet, any software on your computer can use this connection to communicate. Computers 'out there' can try to see if your computer is connected and (using sophisticated techniques) try to take control of your computer.

You will be surprised at how many programs actually try to use the Internet connection – don't worry: most are just checking to see if there are newer versions of the software available. Then you probably have anti-virus and anti-spyware software trying to get updates. The only problem you need to be aware of, is that these updates can be very large to download and can cause your cap to decrease quite dramatically.

Firewall software is software that allows you to control which programs can access the Internet – and tries to hide your computer from the others 'out there' (or at least prevent them from contacting and taking over your computer).

Windows provides a very basic firewall facility. Other examples include Zone Alarm and Norton Internet Security.

Other protection

Make sure that your computer is also protected in terms of updated anti-virus, anti-spyware and anti-adware software. This software is dealt with in module 5.3.

Ongoing maintenance

Okay, so we have the 'house up and going'. What do we need to keep it functioning optimally and to keep it 'clean and tidy' on a regular basis? Think of it as vacuuming the house and putting things away, as well as checking that all the 'windows' stay shut for those nasty burglars (also known as malware). In other words, what kind of management tasks do we need to perform to keep the computer running smoothly and efficiently?

Desktop management

Imagine the desktop as an actual desk where you stack your work, move items around and add and remove items as needed. Shortcuts are often placed on the desktop to provide quick and easy access to frequently-used items like programs, such as anti-virus software, and to specific folders.

Many users, however, develop a bad habit of dumping all sorts of folders and downloads on the desktop for quick access. This can become excessive and you end up with a completely cluttered desktop. It is import to remove seldom-used icons and shortcuts in order to work more effectively. Windows even provides a wizard to guide the user to remove seldom-used items such as icons from the desktop.

File and folder management

In Module 4.2 we dealt with a number of issues around the management of files. Just some quick reminders:

- Keep on refining and maintaining the folder structure that you created. If you find you start having too many files in one folder, it might be an indication that you should 'sub-divide' the folder by adding sub-folders.
- When we are in a hurry, we tend to use default file names such as *Doc1* or *X1*. Locate files that do not have meaningful names and check if you need them still. If you do, rename them with more meaningful names.
- Delete any files that you no longer need.
- You may wish to empty the Recycle Bin which contains all the files that you have deleted from the hard drive to free up disk space. If you have plenty of free disk space you may choose to leave the Recycle Bin in case you need to restore a file you deleted.
- Move (large) files that you seldom use into a compressed file to save on disk space.

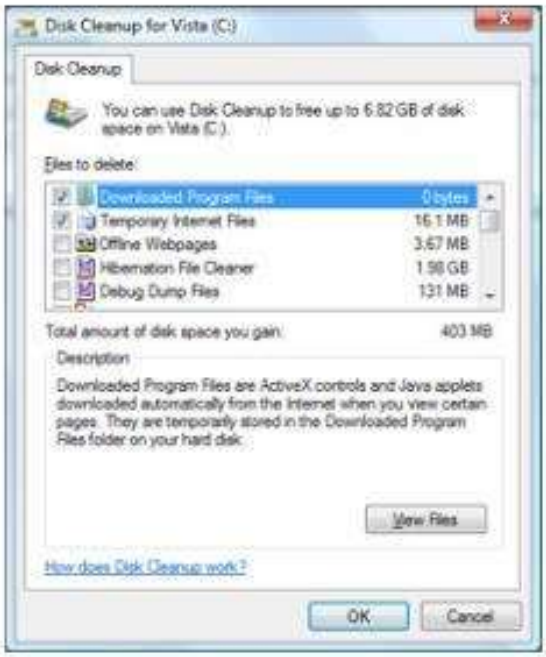
While you are about it, you can also tidy up your e-mail folders using these principles mentioned above. Oh yes, and get all the files you temporarily saved to the desktop cleaned up!

Disk Cleanup

Windows provides a very useful utility, namely the Disk Cleanup Wizard to perform a lot of your housekeeping in one step by removing the following:

- Temporary files downloaded from the Internet.
- Deleted files and folders from the Recycle Bin.
- Temporary files created by Windows.
- Components of Windows that you are not using.
- Installed programs that you no longer or seldom use.

You will be presented with a list of all the different types of files that can be deleted in order to increase the amount of available free disk space.



Backup

One of the most important – and most neglected – maintenance tasks is making backups! Computers can be stolen, fail or be damaged, and files do get deleted – most commonly by user error.

Backing up is simply the process of making duplicate copies of files onto removable media or onto another computer. These should be kept separate from your computer (preferably in a different building) so that you have access to the files in the event of your computer being damaged or stolen.

Windows provides a backup option via the Control Panel. One also gets specialised backup software. Either way, the most important thing is that you do make copies of your work!

Tip

Do yourself a favour and invest in an external hard drive. They simply plug into your USB port and act in the same way as a flash disk from the user's perspective. The amount of data and programs stored on a modern hard drive is enormous – backing up to CD or DVD is becoming impractical and is tedious in comparison.

Check that you can restore files from your backups occasionally, to see that they are actually 'working'.

The process of backing up files has become increasingly difficult with the volume of data that is stored on hard drives these days. The following tips can help when deciding on a backup strategy:

- Make sure you have copies of your original installation disks of your operating system and application programs. We usually only backup our data and not the programs as well.
- Perform the general housekeeping activities that we discussed earlier to make sure that you are not wasting disk space with duplicate or temporary files.
- If you backup at fixed intervals, (say once a week or once a month) you only need to backup those files that were modified since the last backup. Remember that you can search for files modified between certain dates using Windows Explorer.
- Flash disks should preferrably only be used to transfer data from one computer to another and not as backup devices.

You can save space and time by compressing your files. Compressing files decreases their size and reduces the amount of space they use. One or more files can be compressed into a zip file (with a file extension of zip) and files can be extracted from this archive by decompressing them back to their original size so that they can be used again.

Although most operating systems (including Windows) have compression tools, WinZip is probably the most widely used utility program used to compress files. Often you will hear people referring to 'zipping' and 'unzipping', which refers to the compressing and decompressing of files. The process to follow is quite intuitive.

Updates

Virtually all software can be updated via the Internet. Programs need to be updated to add new features which may not have come with the original version

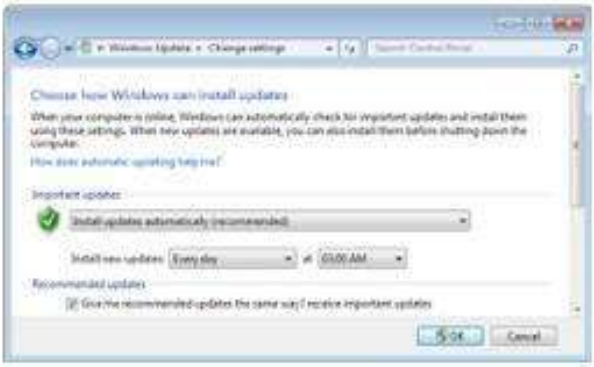
- to fix 'bugs' – errors that exist in the program
- to deal with new threats in the case of anti-virus, anti-spyware and anti-adware software, for example.

Companies often combine all their updates in one unit called a *service pack*.

Anti-virus and anti-spyware software needs to be updated on a regular basis. You can, for example, schedule updates to run at regular, fixed intervals, say on a daily basis.

Firewalls do not have to be updated, as they simply monitor and warn the user if unknown programs are trying to access the computer, or if unknown programs are trying to access the Internet from the user's computer.

In the case of the operating system, it is not as crucial to download the updates as frequently, but some of the updates are termed as 'critical', meaning that they are important to fix. This is often because it has been discovered that hackers or malware can use vulnerability in the software to attack your computer. Windows also allows you to schedule checks for updates to be done automatically. Some people switch off this function, as they may want to choose which updates they want (or they may be running low on their 'cap'!).



Adding and removing software and hardware

In our home we may want to add some things from time to time. In the case of a computer, software and hardware can be added.

Installing software

Programs are either downloaded via the Internet or distributed on CD/DVD. Simply inserting the CD/DVD generally automatically runs the installation and setup program. If this does not happen, consult the documentation that comes with the software. (It usually is a simple task such as running a program called 'install' or 'setup'.)

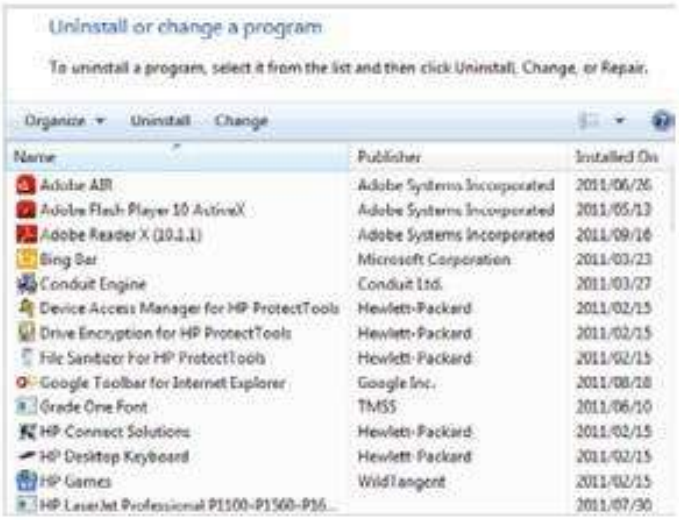
You are guided through the process by an installation wizard – a program that presents you with options and performs the installation based on your choices and preferences. This will vary from program to program but normally includes the following steps:

Agreeing to the licence agreement	Most software can only operate under a strict (legal) license agreement. You normally cannot proceed with the installation until you have read and agreed to the licence agreement.
Inputting a product key or activation code	Most software requires a product code that comes with the software for it to be installed. Alternatively, you may be asked to enter a code to activate the software. Some software will run a few times but then stop functioning until you enter the activation code. This is often e-mailed to you or obtained from the Internet. Its purpose is to reduce illegal installations of the software.
Choosing the folder where the program should be installed	A default folder, where the main part of the program will be installed, will be suggested. This is usually in the Programs folder on the 'C-drive' (hard drive) of your computer.
Choosing the type of installation you want	Normally you can choose between the following types of installation: • Typical – which contains the options and components of the software that most users want. • Custom – which allows the more experienced user to choose the components that must be installed. • Full – where all components of the software are installed.
Installing other extras	Some programs allow you to add other 'useful' software. Be aware that some of these components may be dangerous to install (possible spyware, etc.) unless the software comes from reputable company.
Checking for updates	After the program is installed, you may be asked if you want to check if newer editions of the program are available to download.
Registering online	You may have to register your program online, and sometimes you are asked if want to receive news about future updates, etc.
Adding shortcuts	Most installations ask whether you want the program to be added as a shortcut on your desktop or to the Start menu.
Running the program, or taking an online 'tour' of the program	Finally you may be asked if you want to run the program or take a 'tour' of the program. This often takes the form a video clip or presentation which shows you the latest features of the program, and how to start using it, etc.

Uninstalling software

To remove a program from your computer, you cannot simply delete the folder in which the program is located. The files belonging to the program are often spread across several folders. You may also find that files belonging to other programs can be stored in the same folders, especially the Program Files folder.

Windows provides an option in the Control Panel which allows you to uninstall programs. The uninstall utility program ensures that the program and its associated components are removed correctly.



Adding new hardware

Plug-and-play

Hardware installation used to be a highly technical exercise before computer users' lives were simplified by plug-and-play. Plug-and-play refers to the technology whereby devices that are added or connected to a PC are automatically 'detected' and the PC is configured to use them. Fortunately, the vast majority of these devices connect externally to the PC via a USB port. USB technology is plug-and-play technology, so the devices are automatically detected and the PCs configured, provided that drivers are available for the device.

Drivers

Part of the installation process that occurs with plug-and-play devices, is that a driver is automatically installed. A driver is software that enables the operating system to communicate with a hardware device.

A huge database of drivers comes with the Windows operating system, so in most cases the drives are automatically installed. If a driver cannot be found, you will be prompted to supply a location where the driver can be found, e.g. a CD, or to allow Windows to try and locate the driver on the Internet.

Other housekeeping tasks

There is no hard and fast rule on when you should run your maintenance tasks. Some tasks, such as updating your anti-virus and anti-spyware program, should be done frequently – at least weekly, if not daily, if you are (quite rightfully) paranoid. Some other tasks, such as those described in this section, can be performed as and when needed, but probably only every few months. A lot will depend on your typical use of the computer and local circumstances.

Disk defragmentation

The more you use your computer (adding programs, saving, moving, deleting, creating files), the more the files (and parts of the files) get scattered on the disk. The more 'scattered' the files' parts are, the slower your computer becomes. This is known as disk fragmentation.

A disk defragmenter is a tool that reorganises the parts of files and speeds your computer up again.

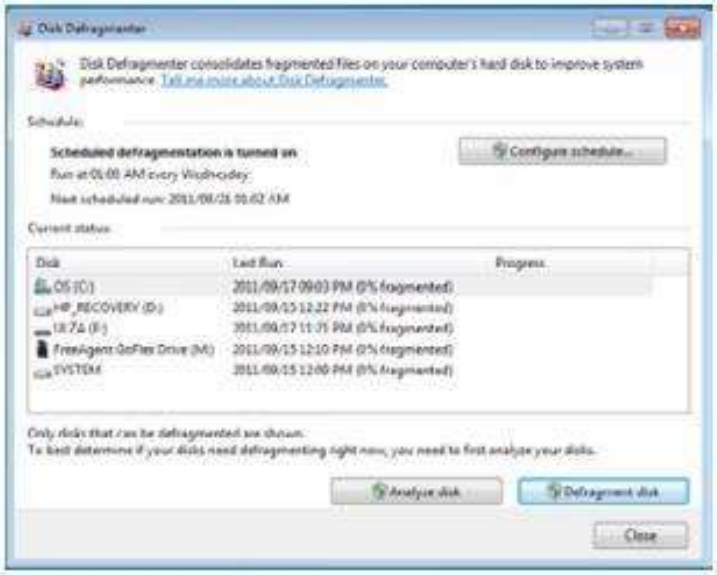
A disk defragmenter is included with Windows but it is possible to buy additional, more powerful versions of the software. For most people the standard version that is supplied with Windows should be sufficient to keep their computer in good condition.

Note that any disk, including a flash disk, where files are saved and deleted will become fragmented at some stage. It is, however, more crucial to defrag your hard drive than a flash disk because of the number of files that are stored on it.

Also note that defragmentation does not free up disk space. It simply re-arranges the files' parts so they are quicker to load from the disk.

When you start the disk defragmenter utility, it will look at your disk and tell you what percentage of the hard drive is fragmented and whether it needs to be defragmented. If the percentage of fragmentation is low, it is not necessary to defragment your disk.

It is, however, recommended that you defrag at least once a month. It is also possible to schedule the running of a defrag operation. The best is to run it at night when you are not using the PC, as it is a fairly time-consuming process, and can slow down access to the hard drive while the defragmentation process is happening.



Archiving

The concept of *backups* and *archives* are often confused. The best way to explain the difference between these two concepts is to look at the aim of each of them. Backups are made so that you have a copy or 'duplicate' of files in case something goes wrong (a file is deleted or the current version is corrupt/overwritten etc.). This is a process that can be automated – you can schedule a backup to run at fixed intervals, and even choose which folders to backup (if not the whole drive).

Archiving refers to storing files that are static (not going to be changed), usually onto a secondary system or onto a DVD for example. For example, let's take a school administration system. You might make a PDF copy of all the reports at the end of the year and store them on DVD as you are not likely to reference them once every pupil has received their report, nor are they going to be changed. In other words, archiving is a 'manual' process where you take files that are not going to change or are likely to be used in the near future, and move them off the main system to free up space.

1 Activity

1. Briefly explain what we mean by housekeeping in terms of computers. Refer to the general aim of housekeeping in your answer.

2. Many people suggest that the Control Panel should be the first place to start when it comes to housekeeping.
   Briefly explain why this is the case by referring to the function of the Control Panel.

3. Your mother often makes use of your father's laptop computer. You suggested that they set up separate user accounts.
   a) Explain what this means and why this is a good idea.
   b) Your mother says that it is not necessary, as she just uses the Administrator account when she needs to work. Explain why this is not a good policy.

4. The network administrator at school has told the grade 9 classes that they will have a new profile the next time they log on. Explain what this means by referring the concept of a profile in your answer.

5. Depending on the restrictions on your PC, you can customise your desktop.
   a) Explain what 'customising your desktop' means by referring to two items you can customise, besides any resolution or monitor settings.
   b) Why are you often prevented from customising your desktop in a network environment?

6. What do we mean when we refer to the concept of the resolution of the monitor?

7. Your mother was outraged when she was told that she should keep her anti-virus software updated 'after she paid good money for it!'
   a) Briefly explain to her why it is essential to keep anti-virus software updated.
   b) Give two other reasons why software should be kept updated.

   Someone recommended to your father that he switch off the firewall on his PC.

8. He went to the Control Panel and located the option to turn the firewall on or off. The operating system prompted him with a message recommending that he should not turn off the firewall.
   a) Explain why it is not a good idea to turn the firewall off, by referring to the function of a firewall.
   b) He also saw an option to 'Block all outgoing connections'.
      Briefly explain what this means, and why this might not necessarily be a good idea.
   c) Is it necessary to keep the firewall software updated? Briefly motivate your answer.

9. One should not use default names when saving a document.
   Explain what is meant by this statement.

10. Your sister empties her Recycle Bin every day when she logs off her PC.
    a) Explain what the Recycle Bin is used for and what it means to 'empty' the Recycle Bin.
    b) What benefit is there in emptying the Recycle Bin?
    c) Describe one potential drawback to emptying the Recycle Bin on a daily basis.

11. Give three ways of regaining free disk space on your hard drive, besides empting the Recycle Bin.

12. Give two reasons why it is important to backup files, besides the fact that they may be accidentally deleted.

13. Your sister has created a separate folder on her hard drive to back up all the files on her hard drive.
    a) Give two reasons why this is not a good strategy.
    b) Outline a better, safer strategy for backing up her files.
    c) Is it necessary for her to back up her operating system and application programs? Briefly motivate your answer.
    d) Give two reasons why you would not recommend that she use a flash disk as a backup device.

e) Besides not backing up the operating system and application programs, outline two ways in which your sister can reduce the number of files she needs to back up.

f) How can your sister reduce the overall size of the backups (not the number of files) she needs to make?

14. One of the reasons why many users do not update their software, is that they simply forget to do it!
Outline a strategy that users can make use of to ensure that the software on their computer is always kept up to date.

15. Your mother brought home a CD to install a new version of the accounting software she uses at work. Her boss told her that it is easy to do, as she only needs to follow the installation wizard.

a) What is an installation wizard?

b) She inserted the CD into the DVD drive but nothing happened. How can she go about trying to install the software?

c) Your mother says that she got asked some question about a licence agreement.
Can she just ignore this? Motivate your answer by referring to the purpose of a licence agreement.

d) The installation wizard prompted your mother to enter her e-mail address so that an activation code can be sent to her.
Explain what an activation code is, and why it is being sent via e-mail.

e) Your mother asked whether she wanted a typical, full or custom installation.
Briefly explain the difference between these three types of installations.

f) The installation wizard prompted her to accept the default folder.
What is the default folder in this context?

g) The wizard asked your mother if she wanted some other free software installed.
Should she accept this offer? Briefly motivate your answer.

h) The wizard also asked your mother whether she wanted to register her product online.
Should she do this? Motivate your answer by referring to the advantages of registering the product.

i) Should your mother want to remove this program at a later stage, can she just delete the default folder that she specified during the Installation process?
Motivate your answer by outlining the correct way of uninstalling software.

16. Connecting additional hardware like a printer to your PC has been made much easier, as most hardware connects to USB ports. In addition, USB technology supports plug-and-play technology.
What do we mean by the term plug-and-play?

17. Your father was told that he does not need a driver for his new printer, as it is plug-and-play.

a) Explain what a driver is, and why this statement is not correct.

b) Where can one find updated drivers for older equipment?

18. Your mother says that her boss at work told her that she should defragment her hard drive and flash drives every day to gain free disk space.

a) Explain what the term disk fragmentation means, and why it occurs.

b) Describe what a defragmentation tool does to solve the problem of disk fragmentation.

c) Comment critically on the advice that your mom's boss gave her.

19. Explain the difference between backups and archives.

Glossary

Administrator account	A special account which allows that account holder (user) to make any changes they wish to the system, including to other accounts
Archiving	To store files that are static (not going to be changed), usually onto a secondary system or onto a DVD for example.
Disk cleanup	A utility that removes: • temporary files downloaded from the Internet, • deleted files and folders permanently by emptying the Recycle Bin • temporary files created by Windows • components of Windows that you are not using • installed programs that you no longer or seldom use.
Disk defragmentation	To reorganise the parts of files and speed your computer up again.
Disk fragmentation	Files and parts of files are scattered on disk.
Drivers	Software that enables the operating system to communicate with a hardware device.
Firewall	Software that allows you to control which programs can access the Internet – and tries to hide your computer from the others 'out there' (or at least prevent them from contacting and taking over your computer).
Installation wizard	A program that presents you with options and performs an installation based on your choices and preferences.
Plug-and-play	The technology whereby devices that are added or connected to a PC are automatically 'detected', and whereby the PC is configured for their use, provided drivers are available for the devices.
WinZip	Probably the most widely-used utility program used to compress files.

MODULE 2.5

Improve your browser

Modular software

Originally, programmers had to try and write software for all possibilities (current and future). This has developed into a nearly impossible mission due to the huge variety in operating systems and devices currently on the market and due to the ever-increasing need for multimedia content on the Web.

Software developers therefore started to develop software so it consists of two parts:

- a 'main' program which will satisfy the needs of most users
- smaller applications (plug-ins) which are enhancements, or which contain specialised functionality.

The use of plug-ins makes programs smaller and easier to distribute via the Internet. The main program contains only the core functionalities. If additional functionalities are needed, the user can simply download and install the appropriate plug-in (also called an add-on). This gives users the option to 'install on demand'.

Let us see what type of functionality can be added to a web browser through the use of plug-ins.

Plug-ins are small applications written to perform specific functions which will enhance application software or which contain specialised functionality. They can be developed by the original creator of the application, or by a third party. Plug-ins are usually free.

Browser plug-ins

A web browser acts as our window on the Internet. These applications enable us to visit a wide variety of websites. The websites we visit contain a variety of multimedia elements. This media is forever developing, attempting to deliver a more visually pleasing and functional product.

Plug-ins are needed when you want to open files such as videos, sound or even documents directly in the browser – i.e. when you do not want to download the file and open it in another program.

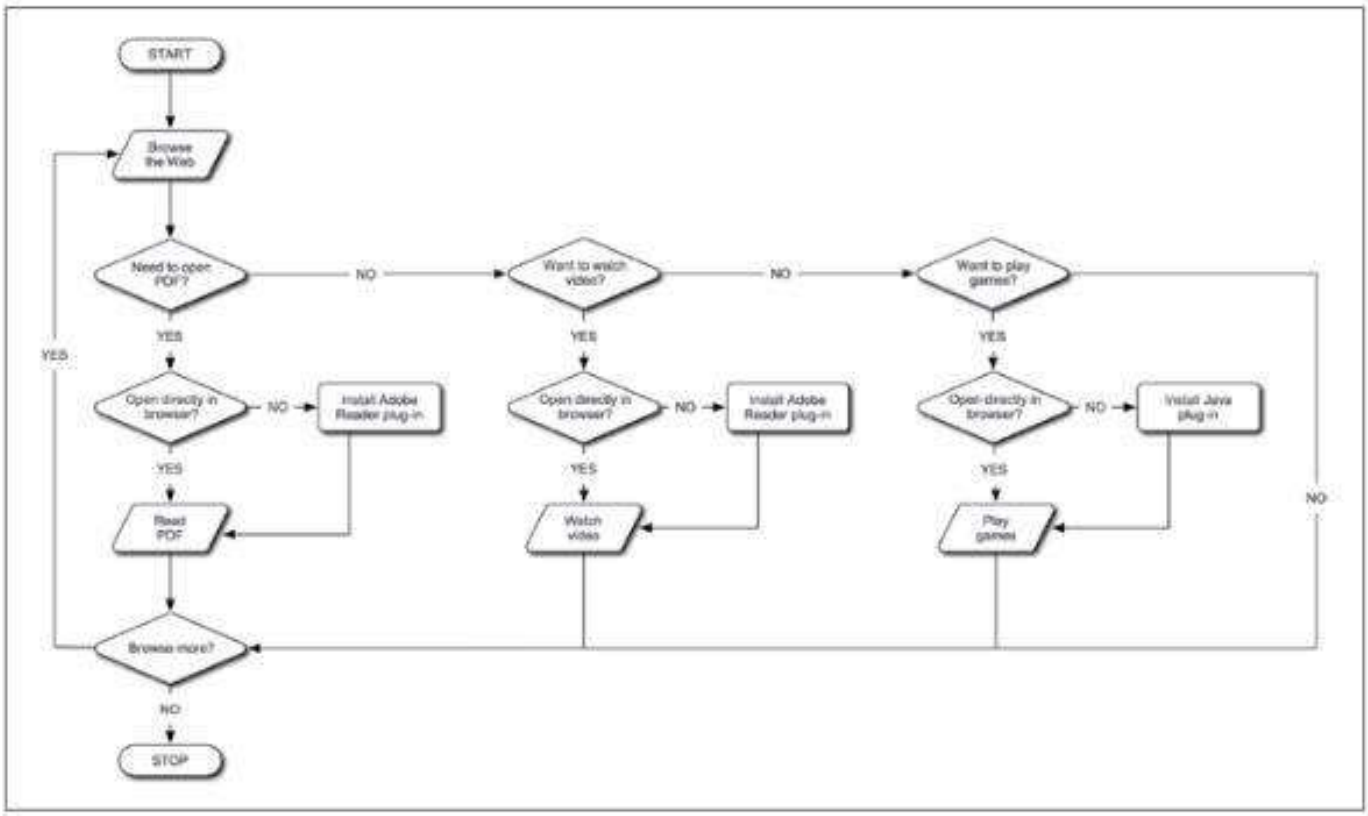
Take a look at the flowchart on the following page. This depicts typical situations where you will need to install plug-ins.

Typical plug-ins for browsers include the following:

- **Adobe Reader plug-in**

  The most widely accepted format for distributing electronic documents is the Portable Document Format (PDF).

  When you install Acrobat or Adobe Reader on a system with a browser, the browser is automatically configured to open PDF files within the browser window – otherwise you will need a PDF reader such as the Adobe Reader plug-in.

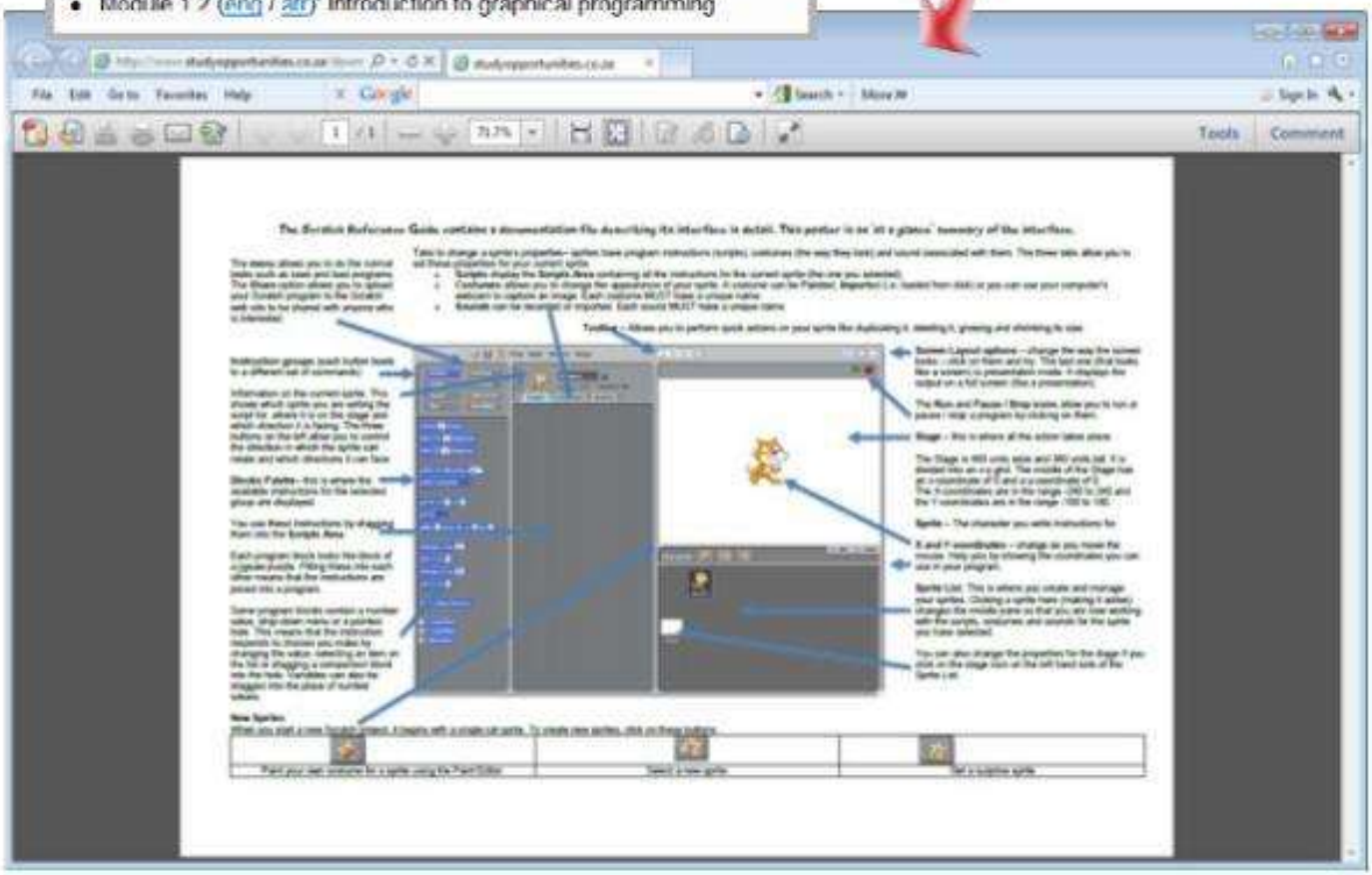


Suppose the following is a section that appears on a website:

Samples from new IT Gr 10 textbook: IT is gr8! @ Grade 10

- Labelled Scratch Interface (eng / afr)
- Link for one of the videos included on the DVD (with teacher's guide). http://www.youtube.com/watch?v=jT8-f0H3GgA
- Module 1.2 (eng / afr): Introduction to graphical programming

A document containing a labelled Scratch Interface is available as a pdf file. If you have the above-mentioned plug-in, and click on the link, it will open within the browser itself.



- **Adobe Flash Player plug-in**

  This is a multimedia plug-in which enables users to experience a very wide range of applications and videos on any browser, irrespective of your operating system.

- **Adobe Shockwave Player**

  Shockwave Player is used to display Shockwave encoded content. It is mostly used for interactive training and multimedia product demos. Shockwave player is also used for media rich and multiple user online games.

- **Google Earth plug-in**

  Browse and see the sights on a 3D globe. The plug-in is also used in Google Maps.

- **Java**

  This is a runtime environment which allows you to run Java applications. We often find that plug-ins for other applications are written in Java. Examples include online calculators, chat interfaces, image viewers, online games, etc.

- **QuickTime**

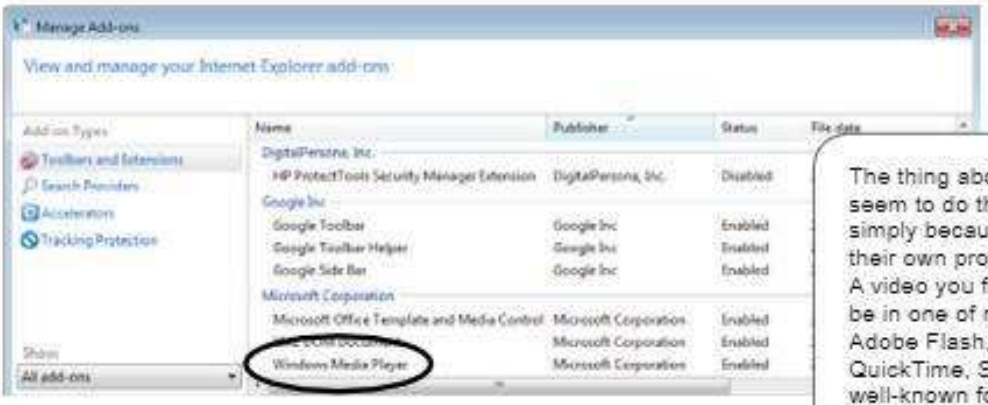
  View video, music and virtual reality elements.

- **Silverlight**

  Silverlight is meant to handle interactive applications and online video. It was created in late 2007 and is growing in popularity. It provides abilities like animation, web services and games, and it also runs on mobile devices.

- **Windows Media Player plug-in**

  With this plug-in for your browser, you can play embedded audio and video on web pages without opening the Windows Media Player application.



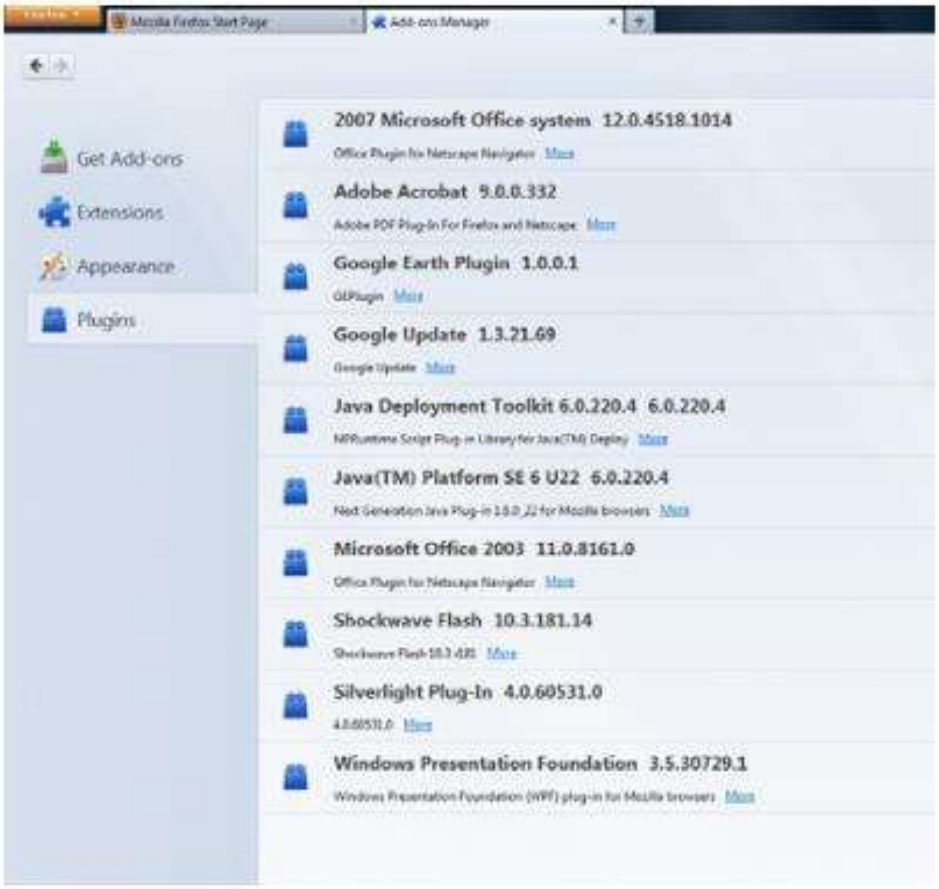
The thing about plug-ins is that many seem to do the same thing – that's simply because different companies use their own proprietary format for content. A video you find on the Web might be in one of many formats, such as Adobe Flash, Windows Media, Apple's QuickTime, Silverlight or another less well-known formats. Each one of these formats will need its own plug-in to allow a browser to play the video. Different sites will choose different formats, so in the end you will need to install the plug-ins for all formats.

1 Activity

1. While viewing a page on Facebook, you are asked to install a plug-in to view the full contents on the page.
   a) What is a plug-in, and why would you need it on a Facebook page?
   b) What are the benefits of plug-ins, from a programmer's point of view?
   c) In which programming language are many of these add-ons or plug-in applications written?

2. A friend wanted to download an application form for University, but a pop-up appeared, asking her to make sure that she had a PDF reader installed.
   a) What is a PDF document?
   b) What add-on would she need to read and print this application form?

3. Consider the following screenshot of a Firefox browser, showing the plug-ins installed on the system:



   a) What would the Shockwave plug-in be used for?
   b) What would the Google Earth plug-in be used for?
   c) What would the Silverlight plug-in be used for?

Glossary

Adobe Reader plug-in	This plug-in allows you to view and print the contents of PDF documents directly from within your browser.
Flash Players	Examples include Adobe Flash Player, Silverlight and Adobe Shockwave Player. These multimedia plug-ins enable users to experience a very wide range of applications and multimedia web content in any browser, irrespective of the operating system.
Java	Generally used to write plug-in programs for other applications.
Media Player plug-ins	A plug-in such as QuickTime, which allows you to play embedded audio and video on web pages without needing to have or open the Windows Media Player application on your computer.
PDF	Portable Document Format (PDF). A popular format to distribute documents electronically.
Plug-in	Software which adds functionality to a bigger program, often used when browsing the Web.

Section 3

Communication and Internet Technologies

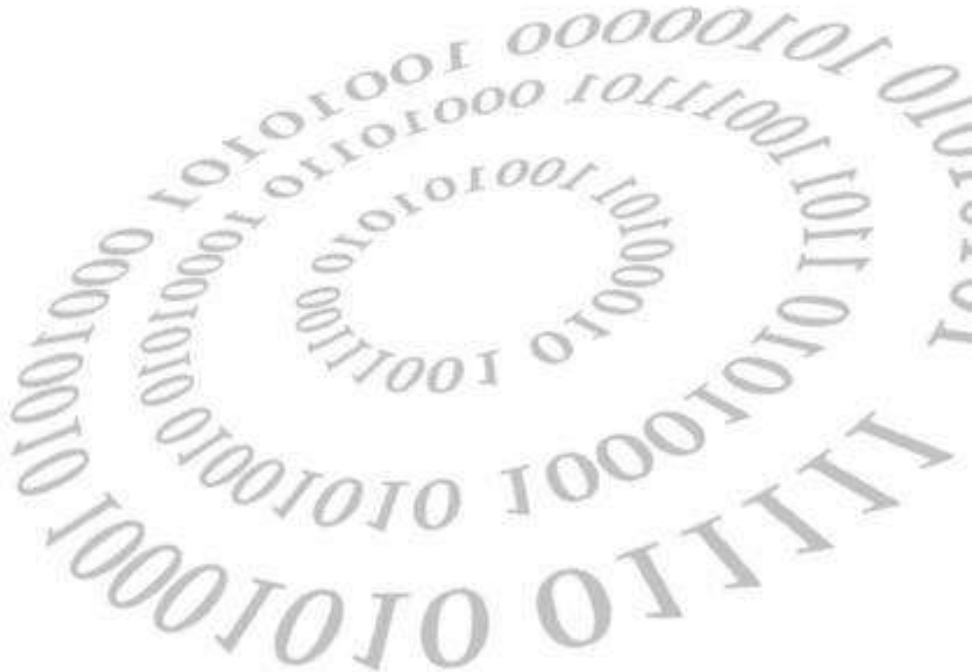
Modules

Module 3.1 Networks

Module 3.2 Electronic communication

Module 3.3 Internet and WWW

Module 3.4 Web page technology

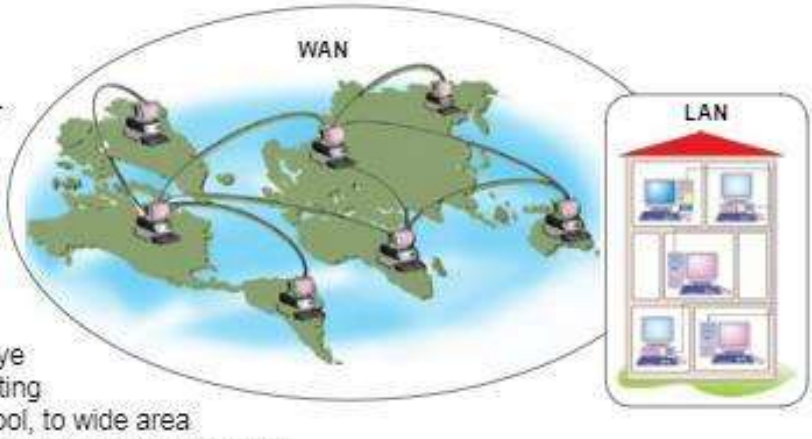


MODULE 3.1

Networks

Connecting devices in a network brought a whole new dimension to the use of computers and technology. Very few computers are not networked in some way these days. It is therefore important to understand the basic concepts associated with networks, as well the distinct benefits they bring in terms of e-communication as well as sharing of resources and information, to name but a few.

What is a network?



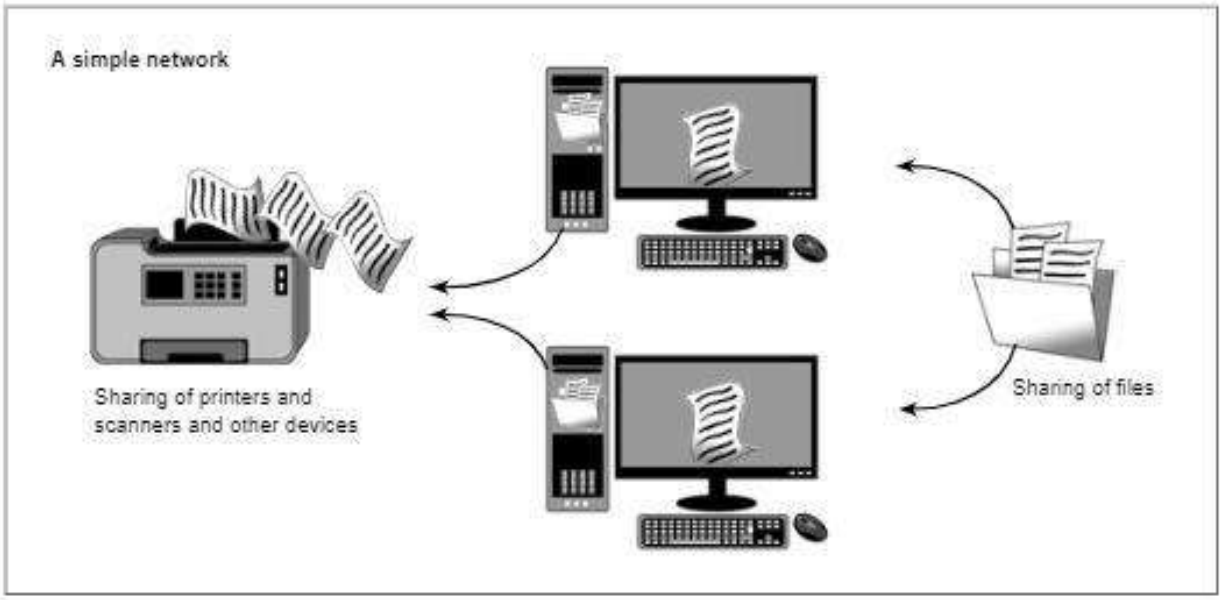
A **network** is a collection of computers or other computing devices such as smart phones that are connected by some sort of communication media (either via cable or wirelessly) to allow users to share hardware, software, data and information, and to communicate.

Modern networks have evolved over the ye from local area networks (LANs), connecting computers in a small area such as a school, to wide area networks (WANs) which are spread over wide geographical areas.

As the need for mobility and flexibility has arisen, networks now even include using our cellphones to check e-mail, surf the Net, access social networks and much more.

Reasons for having a network

Having a network is all about convenience, increased productivity and communication. Explanations of some of the different ways in which networking is used today follows:



Fast, efficient communication

When using a network, people are able to communicate easily and efficiently with each other using amongst others e-mail, chat and video conferencing.

- In a school situation, **e-mail** or SMSs can be used to allow teachers to communicate with parents. For example, the school can send an e-mail or SMS to all the families informing them of important events or when reports are going to be issued.
- **Chat** is an application that gives two or more people the ability to have a 'conversation' by typing on a computer or cellphone. The chatting people can be in the same town, country or across the world and are connected via the Internet.
- **Video conferencing** is an application that allows two or more people to hold on-line conferences at different locations using a computer network to transmit audio and video data as if they were all in the same room.

A printer being shared in a school network

Sharing of hardware resources

Equipment such as disk drives, printers, DVD players, scanners and plotters can be shared amongst many users on the network. This is a lot cheaper than buying such devices for each computer.

Centralisation of data

In a business or home office situation, often more than one person needs access to the same information. By networking your computers you can allow other users to access and use files or folders on a central computer (or on your computer) as if the file or folder was actually on their own computer. It also becomes easier to maintain the integrity of data (to prevent that it becomes out-dated and unreliable) as it is only updated in one central location. You get the benefit of reducing the likelihood of having duplicate or out-dated files on different machines.

For example, a school administration system can be stored on the file server and teachers can then enter their marks from many computers in (or even outside) the school on the same database. This ensures that the data is the same for all the users on the network.

Transfer of files

Networks allow people to transfer files without the limitations of using flash drives or CDs from computer to computer, whether the computers are in the same office or across the world. The files can be transferred using the network.

For example, in a school environment, a teacher can save a file on the server and all pupils using the network can access that file and copy it to their directories or computers, or the teacher can retrieve all their files after a test and move them to her directory so that they can be marked.

Leisure

'LAN parties' (LANing) are popular activities where people get together and play games on a network. Even though they're called LAN parties, they don't always stop at local area networks, sometimes including people across the world using the Internet. LAN competitions have also become popular where teams from different schools compete against one another.

Newer home entertainment products such as digital video recorders (DVRs) and video game consoles now support either wired or wireless home networking. Linking these devices to your home network allows online Internet gaming and video sharing.

Increased control and security

The data of an organisation can be centrally stored on one or more computers. It is therefore easier to ensure that the data is kept secure and backed up on a regular basis. It is also easier to control who has access to the data. Software such as antivirus software can be automatically updated using the network.

Flexible access

In a network, users can access their information from any computer on the network. Many users can collaborate on the same project using the network. If you have an Internet connection to your network, you can work on the same data from anywhere in the world.

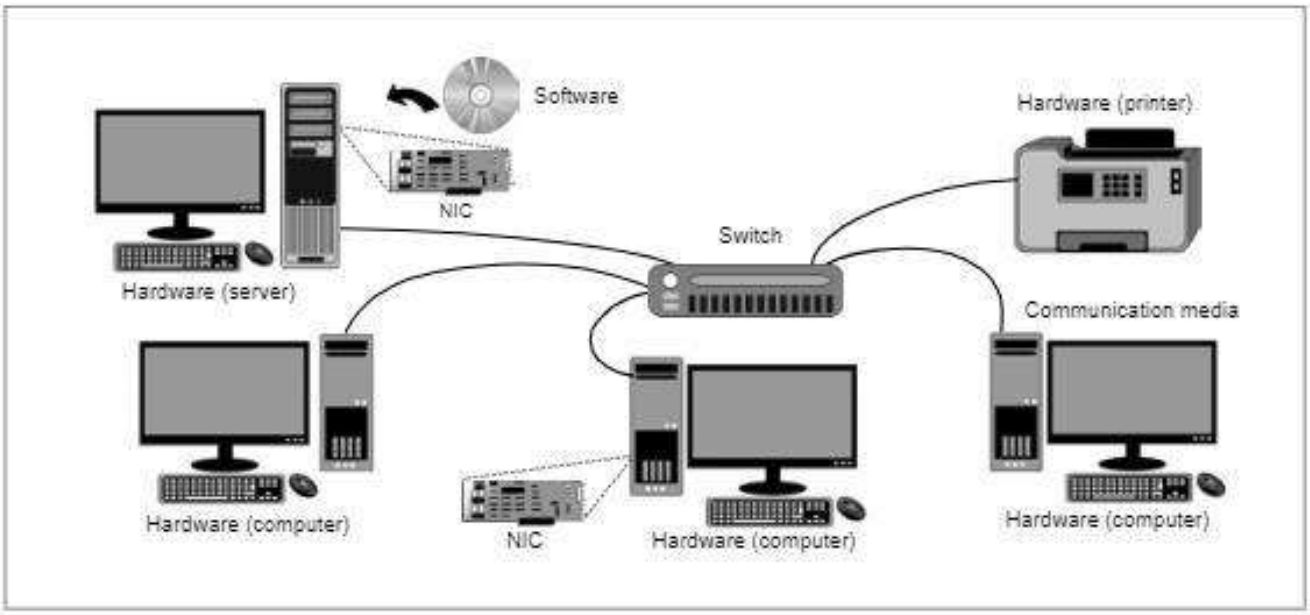
Disadvantages of having a network

Today, most organisations can no longer function without a network. As with most things in life, there are also disadvantages to having a network.

- A higher level of expertise is necessary to manage a network and an organisation will have to hire a network administrator to manage the network.
- A high level of security needs to be set up to ensure that information is kept confidential and data cannot be stolen. Folder access needs to be strictly controlled.
- Malware such as viruses can spread quickly via a network.
- If the network develops a problem, potentially no users can work, print or access their data.
- A central fault in the network such as the server crashing can cause a loss of data.

Components of a network

A network consists of the following basic components:



Hardware

Computers and peripherals

Hardware in a network includes the computers that are linked in the network as well as other hardware that can be shared such as printers and scanners.

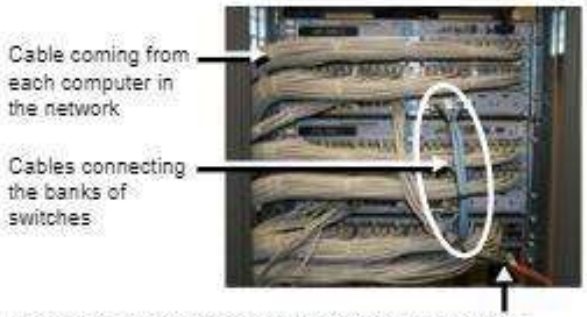
A server is a computer that provides shared resources, such as files, e-mail and Internet facilities or printers to network users. The most common type of server is a file server which is used to store all the users' files instead of each user saving their work on the hard drives of their own computers. In this way users can access their files from any computer connected to the network and they do not have to worry about the backing up of files themselves. A server is not used as a 'user' computer and will usually have better hardware specifications such as more memory (RAM) and large hard drive space.

Network interface controller

Every computer in a network has to have a network interface controller (NIC) to link the network cable to the computer and to allow communication of the computer with the network. It converts data sent from the computer into a format that can be sent across the network. Only the edge of the interface can be seen at the back of the computer and has a port into which the cable gets plugged. Network interface controllers are built into the motherboard of all computers. If the on-board controller gets damaged in any way, then a separate network interface controller can be plugged into an expansion slot on the motherboard.

Switch

A switch is a hardware device used to connect computers on a network so that communication can occur. One end of the network cable plugs into the network interface controller of the computer and the other plugs into a port on the switch. The switch can then send the data received from one computer to the specific computer(s) that it must be sent to and manages the communication on the network efficiently.



The specification 10/100/1000 often seen in computer adverts refers to the data transfer rate (speed) in a network i.e. 10 Mbps / 100 Mbps / 1000 Mbps. This is controlled by the switches and NIC.

A large network belonging to an organisation will have many switches connecting all the computers in the network. They will be stored in a cabinet such as the one shown in the photo. Notice how many cables are connected to each switch.

Communication media

Computers in a network have to be connected by some sort of communication media. While wireless connections using radio, microwave or satellite are common for WANs, many LANs today still utilise cables of some sort to allow computers to communicate.

Wired

Cables are the most common way of connecting computers in a LAN. The most common cable used is UTP. Although the price of fibre optic has decreased dramatically and has become more affordable for use in a LAN to connect the different computers, wireless options and gigabit UTP have become more popular.

Fibre is now mainly used as a backbone (a cable connecting different LANs). For example, if a school had a computer centre using UTP and they also had a LAN in the administration section using UTP, they could use fibre to connect the two.

- **Unshielded Twisted Pair (UTP)**

  UTP consists of one or more pairs of copper wires twisted around each other forming the cable. Unshielded refers to the fact that the wires are not covered in a foil insulating layer for protection.

  UTP looks just like normal telephone cable, but it is of a higher quality necessary for carrying data.

  Some important characteristics include:

  - The connector type for plugging the cable into the computer or switch is called a RJ45 connector.
  - UTP is easy and inexpensive to install.
  - The data rate is normally 100 Mbps but gigabit data rate is also available.

UTP cable plugs into the network port on a PC

- **Fibre optic cables**

  Fibre optic cables consist of many very thin strands of glass or plastic each surrounded by cladding (to prevent the light from leaking out of the fibre) and then covered by a final protective outer cover.

  Fibre optic uses light beams rather than electrical signals to transmit data. Lasers are used to generate these beams of light. The presence of a pulse of light represents a one (1) and no pulse of light represents a zero (0).

  Some important characteristics include:

  - Fibre is thin and lightweight; therefore it can be used in cramped areas.
  - Relatively complex to install and configure. Each connection must be made carefully so that path of light is not obstructed. With fibre optic cables, sharp corners cannot be negotiated. It can be easily damaged, so it has to be installed in armoured casing. If a section of the cable becomes damaged, the whole cable has to be replaced, as it cannot be repaired.
  - Fibre supports very high transmission rates.
  - Network interfaces, cables and other devices that are used together with this type of cable are relatively expensive.


Fibre optic cable with connector

Weaknesses of communication media Communication media have specific strengths and weaknesses based on physical characteristics.		
Attenuation	Attenuation refers to the loss of signal strength over a distance. The length of a cable can affect the amount of attenuation. With a **wireless** network, access points have to be placed strategically so that the area of communication is covered sufficiently.	
	UTP is susceptible to attenuation therefore the maximum cable length that can be used without having to boost the signal is 100 m.	**Fibre** has much lower attenuation than UTP. It can transmit over distances measured in kilometres.
Eavesdropping	Eavesdropping is related to EMI. Since a cable is susceptible to EMI, it then becomes possible for someone to detect a signal on a cable without piercing the cable and so gain access to the data. This is a problem if a network needs a high level of security. This is also a concern with a **wireless** network.	
	UTP is sensitive to eavesdropping although the twisting does help reduce the sensitivity.	**Fibre** is immune to eavesdropping. If the signal was tapped, the flow of light would be broken and the tapping would be detected.
Crosstalk	The magnetic fields in two wires that are close to one another can interfere with transmission and create crosstalk. This can result in the loss or corruption of data.	
Electromagnetic interference	EMI is an electronic signal that interferes with normal network transmission. Possible sources include engines, machinery, lights, tools and radar. Shielding, as well as avoiding placing cables near to EMI sources, can reduce this. **Wireless** networks are also affected by EMI.	
	UTP is sensitive to EMI and power surges although the twisting does help reduce the sensitivity.	**Fibre** is not affected by electromagnetic interference and power surges. It can be used in areas such as factories and oil refineries where electromagnetic fields can affect communication.

Wireless

When a wireless network is installed, it usually has wireless access points that allow radio communication between devices. These access points are in turn still connected to a cabled network for access to the server and other devices. If a wireless network were to be installed in a school for example, access points would have to be installed around the school which send out the radio waves that allow the communication to occur. The number of access points will be determined by a number of factors including the size of the school as well as the distances between buildings.

The computers or devices that connect to the network need to be wireless enabled or have a wireless network controller installed.

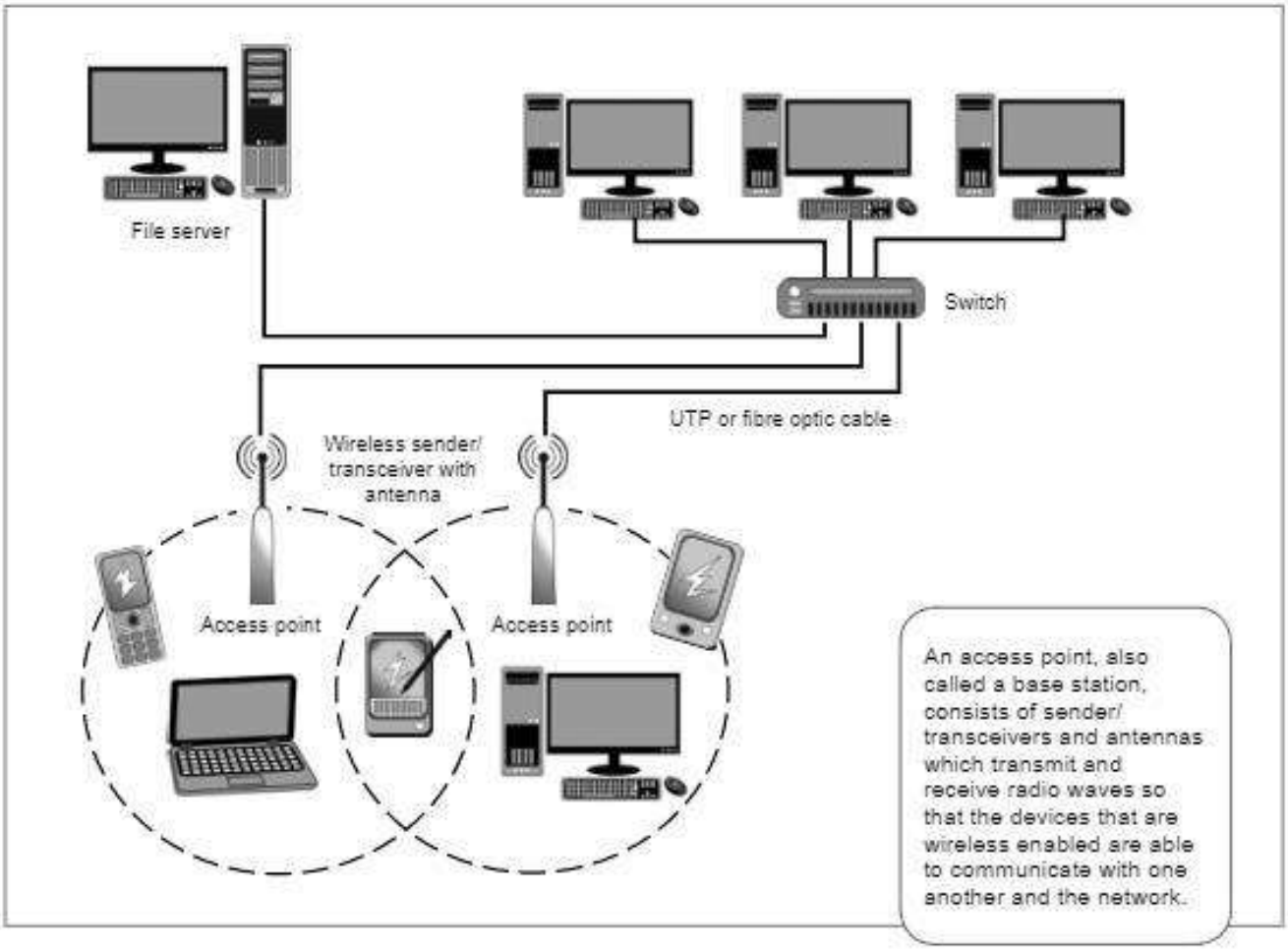
Network operating system

A network operating system is software that controls all the communication in a network as well as the security on a network. Most operating systems such as Windows 7.0 have built-in networking capabilities. For larger networks, specialised 'Server' editions of the operating system need to be installed on the servers.

Types of networks

Local area network (LAN)

A LAN is a network connecting computers in a small area such as a school, university campus or company offices in one building. The computers are connected either using cables, or wirelessly. This has given rise to another classification called a WLAN or Wireless LAN. In other words it is still a LAN but it is implemented using high-frequency radio to communicate instead of cables.



Advantages of a WLAN

- It is easier to add or move computers as the network points are not fixed – no cabling is required.
- It is easier to provide network connectivity in areas where it is difficult to lay a cable such as in historical buildings and factories.
- Installation is far easier than with a cabled LAN as cables do not need to be pulled through walls or ceilings.
- Adds mobility for laptop users.

Disadvantages of a WLAN

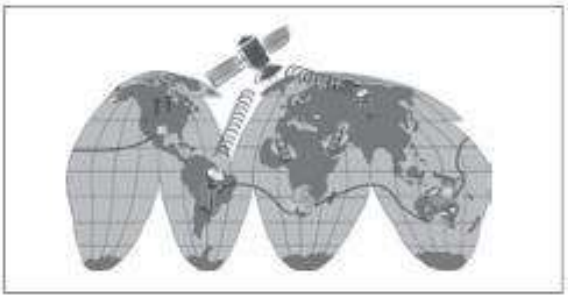
- Security can be compromised unless the WLAN is configured correctly to ensure that the connection cannot be hacked. Unauthorised users may also make use of your Internet connection, costing you money.
- Performance can decrease as the number of computers connected increases.
- Signal could get lost due to electronic or electrical interference.

Wide area network (WAN)

A WAN is a network that is spread over a wide geographical area, such as a city, across provinces or countries and even continents. A WAN is often used to link groups of LANs. Communication takes place by means of telephone lines, satellite or microwave.

Examples

- The cellphone network linking different areas in South Africa is an example of such a network. When you drive around, look for the cellphone towers that are used for cellphone communication.



- The Internet is another example of a WAN. The computers accessing the Internet use a combination of cables, microwaves and satellite to communicate. Some people in fact refer to the Internet as a global area network or GAN as it spans the whole world.
- The network of a bank which has offices and ATMs in all cities in South Africa, is yet another example of a WAN.

Intranets and extranets

An intranet is an organisation's private network and is an Internet-like environment consisting of web pages relating to the organisation's business. The information is only available to the organisation's employees. If access to other users, outside the organisation, is provided, then this is called an extranet. Access will be granted using a username and password. For example, an extranet can be used to grant an organisation's clients access to certain areas of the organisation's intranet.

1 Activity

1. Distinguish between a LAN and a WAN.
2. Name and briefly discuss three reasons for having a network.
3. Discuss two possible disadvantages of using a network, besides any issues relating to the Internet.
4. What important item does a computer need to have in order for it to be connected in a network?
   What is its function?
5. Many local area networks are now set up using wireless technology.
   State one advantage and one disadvantage of this approach.
6. Will it be possible to get access to information on an organisation's intranet?
   Motivate your answer by explaining the difference between an intranet and an extranet.
7. Some friends want to organise a LAN party in the school hall on Saturday evenings.
   Describe what hardware and software would be needed to implement this 'party'?
8. Briefly explain what the term WLAN means.
   What would you consider to be the main benefit of implementing a WLAN?
9. Describe three of the major drawbacks of using cabling in a network.
10. Fibre optic cable is often used as a backbone in networks. Explain what this statements means and motivate why fibre optic cable is the better solution in this circumstance.
11. How does a server differ from other computers in a network?
12. What is the main function of a switch in a network?
13. How can having a network simplify backup strategies and contribute to avoiding unnecessary duplication of data?

Client-server vs. peer-to-peer LAN

The network operating system controls the information flow through the network and determines the roles of the computers in the network.

The computers in a LAN can be classified according to the services they provide. Computers in a LAN have three main roles, namely clients, servers and peers.

Clients

Clients are computers that use but do not provide network resources. They rely on servers for resources such as files and printing.

- *Fat clients* run most of their applications from their local hard drives and make little use of network services.
- *Thin clients* have low specifications, and depend almost completely on a server to run applications and store data files.

Servers

Servers are computers that only provide network resources. They are powerful computers with better, hardware specifications such as more memory and larger hard drives and are never used as normal users' PCs.

The functions of a file server include:

- Manage data on the server's hard drives and network traffic.
- Control access to its services by other PCs in the network.
- Provide users with access to the files that are stored on it.

While the file server is the most common server found in a network environment, there are other types of servers.

- E-mail servers – used to handle all the e-mail needs of the organisation.
- Internet or proxy server – users connect to Internet via this server so security control for viruses etc. can be implemented at one central point. Access to the Internet can also be speeded up by storing (caching) recently accessed web pages on disk.
- Print servers are used to handle all the printing requirements of a network with a large number of computers and printers.

Peers

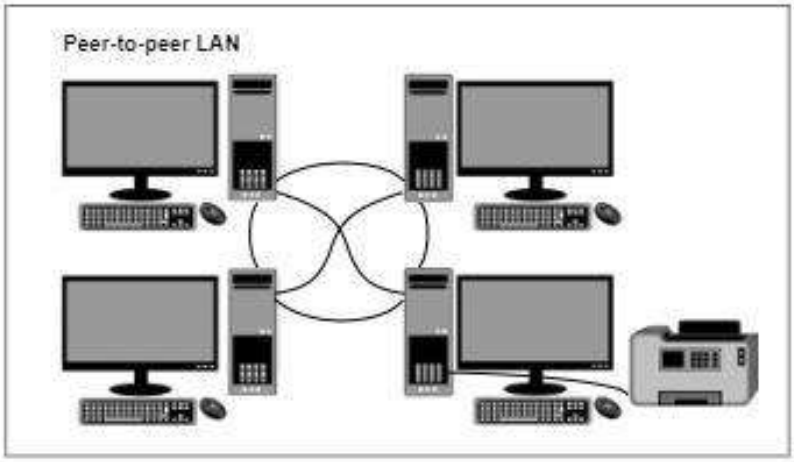
Peers are those computers that both use and provide network resources.

The way that information is controlled in a network classifies LANs in two formats, namely peer-to-peer and client-server.

Peer-to-peer LAN

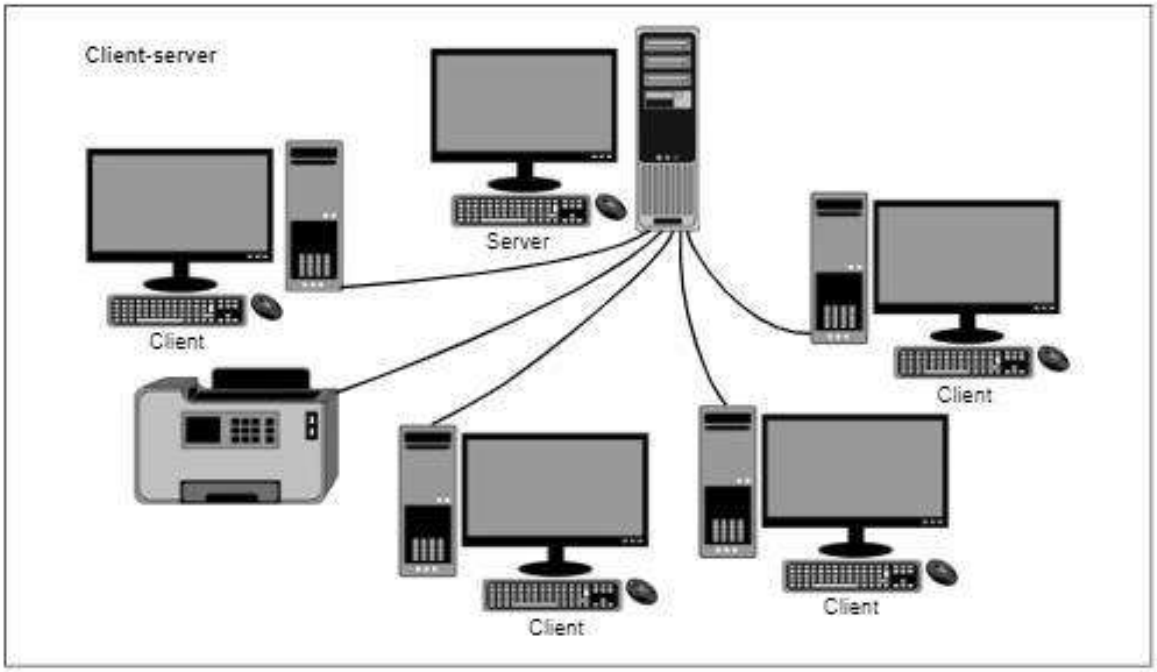
A peer-to-peer network is a local area network in which network resources are shared among workstations, without a dedicated server. It is a simple, inexpensive network that usually connects ten or less computers.

Peer-to-peer LANs are normally used to link computers in a home or very small business so that all members are able to access the information on each computer and to share printers and Internet connections.



Client-server LAN

In this system, each computer is either a client or a server. The main computer is called a server. The PCs on the network are called clients. A client-server LAN provides an efficient way in which to connect many computers.



Comparison	
Peer-to-Peer	**Client-Server**
All computers are peers (equals) and perform the same communication functions.	Each computer in the network is either a client or a server.
No server is necessary.	A server is required.
All the computers have high specifications with large disk capacity and powerful processors.	The rest of the computers in the network do not have to have a large disk capacity and can have a less powerful processor. Processing is still local.
Software is cheaper as it is often built into many operating systems, for example Windows 7.	A specialised network operating system such as Windows 2008 Server edition has to be installed on the server side.
Can only be used with a small number of computers. Suitable for linking a few computers in a home or small office environment.	More suitable for a larger number of computers.
Can be installed by a person with reasonably low technical skills.	Has to be installed by a person with high technical skills such as a network engineer.
Can have a slow performance.	Faster performance.
Security is limited.	Security is far more sophisticated.
Since there is no server, the machines can run as standalone computers should a network fault occur.	If the server crashes, or if there is a cable fault, network dependent computers cannot be used.
Does not need a dedicated network administrator.	Needs a dedicated network administrator.
The operating system that is loaded on each computer allows the sharing of files and resources.	The network operating system is loaded on the server and client software is loaded on each computer.

Access Control when logging into a network

When working in a network environment, it is very important that access to data in the network is controlled. For example, an organisation would not want all employees to have access to files to do with salaries. In a school situation, for example, it is important that pupils do not have access to tests and exams before they are written or that they are not able to change their term's results.

When working in a network environment, you use a username and password to gain access to the network and its resources. This is necessary to control access to and secure data in the network.

Network security refers to policies put in place to ensure the security of a network by preventing unauthorised access and misuse of the computer network.

You will be issued with a username according to the policies set up by the organisation. You will then need to choose a password that will only be known to you. It is important that you keep your password secret as you will be held responsible for any actions done on the network under your username and password.

The following are a few rules to follow when choosing a password that will be difficult to guess or crack:

- Choose a password that is at least 8 characters long. In fact the longer the password the harder it is to crack.
- The password should contain a mixture of upper and lowercase letters, numbers and special characters.
- You should not use your name, you girlfriend or boyfriend's name, your pet's name or anything else that is easy to guess about you. In fact meaningless passwords are the best.
- You should change your password at least every 2 months. In fact most network operating systems have a function where it can be set to force a password change according to a set interval.
- Be careful of using passwords that follow a pattern on the keyboard such as 'asdf' or '1234' as these are also easy to guess.
- Try not to repeat characters for example qq11.
- Be creative!

The following would be an example of a 'good', secure password: q&F1oP%*>41z

How do you know that you are connected to a network?
The following are indicators:

- Presence of extra (logical) drives such as U, V and T which are simply folders on a server that are appearing and acting as drives.
- Presence of other computers in My Network Places.
- Access to devices such as printers not directly attached to the computer is possible.
- Indicator showing network activity in the system tray.

2 Activity

1. Give two ways that a user might notice or check to see if his or her computer is connected to a network.
2. Give three tips for setting up a secure password.
3. A company needs to decide on whether they should install a peer-to-peer network or a client-server network.
   a) Discuss four main features of a client-server network that makes it different from a peer-to-peer network.
   b) The company has chosen a client-server network. Give two possible reasons for their choice.
4. Computers in a network have three main roles - client, server and peer. Discuss the main difference between these three.
5. Discuss two advantages of having a peer-to-peer network over a client-server network.

6. Discuss two disadvantages of having a client-server network over a peer-to-peer network.
7. Thin client solutions have become increasingly popular in recent years.
   a) What does 'thin client network' mean?
   b) Describe a thin client.
   c) Discuss the advantages of such a solution.
   d) Discuss the disadvantages of such a solution.
8. Consider the statement: "There is only one type of server and that is one that is a computer that provides network resources."
   Discuss whether this statement is correct or not and giving other examples if necessary.

Glossary

Clients	Clients are computers that use but do not provide network resources.
E-mail servers	E-mail servers are used to handle all the e-mail needs of the organisation.
Extranet	An extranet is a network where access is provided to the organisations' private network to users from outside the organisation.
Fat clients	Fat clients run most of their applications from their local hard drives and make little use of network services.
Internet or proxy server	Users connect to the Internet via this server so security control for viruses etc. can be implemented at one central point. Access to the Internet can also be speeded up by storing (caching) recently accessed web pages on disk.
Intranet	An intranet is an organisation's private network and is an Internet-like environment consisting of web pages relating to the organisation's business.
Local area network (LAN)	A local area network is a privately owned network connecting computers in a small area such as a school, university campus or company offices in one building.
Network	A network is a collection of computers or other computing devices such as smartphones that are connected by some sort of communication media (either via cable or wirelessly) to allow users to share software, hardware, data and information.
Network operating system	Network operating system is software that controls all the communication in a network as well as the security on a network.
Network security	Network security refers to policies put in place to ensure the security of a network by preventing unauthorised access and misuse of the computer network.
Peers	Peers are those computers that both use and provide network resources.
Peer-to-peer network	A peer-to-peer network is a local area network in which network resources are shared among workstations, without a dedicated server.
Print servers	Print servers are used to handle all the printing requirements of a network with a large number of computers.
Server	A server is a computer that provides shared resources, such as files, e-mail and Internet facilities or printers to network users.
Switch	A switch is a hardware device used to connect computers on a network so that communication can occur.
Thin clients	Thin clients have no hard drives, but depend almost completely on a server to run applications and store data files.
Wide area network (WAN)	A WAN is a network that is spread over a wide geographical area, such as a city, across provinces or countries and even continents.
Wireless LAN (WLAN)	A wireless LAN (WLAN) is a LAN which communicates using high-frequency radio waves rather than cables to communicate.

MODULE 3.2

Electronic Communication

E-communication refers to all forms of communication via electronic means. In other words, it is the process by which computers and other communication devices such as cellphones and smartphones are used to exchange messages and/or digital data.

Applications to facilitate e-communication

There are many applications or tools (programs) available to facilitate e-communication. These are evolving and changing at a rapid rate as the technology improves and changes. These applications can be used with people in the same town, across the country or even across the world. A few of these applications are discussed below.

E-mail

E-mail (electronic mail) is the composing, sending and receiving of messages electronically over a network including the Internet. E-mail will be discussed in greater detail later in this module.

Social networking sites

A social networking site is where individuals (or organisations) can post their likes, dislikes, interests, photographs and activities for their friends (and virtually everyone else unless access is blocked) to see and read.

The idea is that the social networking site allows people to keep in touch with each other even if they are far apart or do not have time to meet in real-life. They can also do all sorts of 'fun stuff' and even meet and make new friends with people they might never meet under other circumstances. Some social networking sites are more business oriented, to allow business people to make and maintain contacts.

Not all social networking sites offer the same facilities or features but they are all generally focussed on sharing information and keeping contact with other users. Examples include Facebook, Bebo, MySpace, Linkedin, Flickr.

Many schools and companies block social networking sites to avoid so called 'cyber-slacking' where students or employees spend too much time on these sites instead of working. Other reasons include the abuse of resources, such as bandwidth, especially by downloading and uploading of large files such as videos and photographs. Companies might also restrict social networking sites so that employees do not transgress company policy and ethical codes by discussing the development of products or simply gossiping about fellow colleagues online.

Online chat

Online chat is an application that gives two or more people, often strangers, the ability to have a 'conversation' by either typing on a computer and using the Internet or chatting by typing on a cellphone. This 'conversation' happens in real-time and the text appears for the other person as soon as 'enter' is pressed.

A chat room is a website that allows people to meet and chat with other people with common interests in real-time. MixIT is very popular as a cheap cellphone chat application.

Instant Messaging (IM)

IM is a live text-based communication facility usually used between two people using computers or other devices such as cellphones. More sophisticated IM software allows voice and video communication. Unlike the more general online chat where you can communicate with anyone who joins the 'conversation', IM communication occurs between 'buddy lists' meaning that you have to know and invite the person you are talking to. It is different to e-mail in that it is often faster as it happens in real-time ('live'). IM facilities are built into many social networking facilities such as Facebook.

Examples of IM applications include Yahoo Messenger, Windows Messenger and Google Talk.

Messaging with SMS and MMS

Messaging refers to the exchange of brief typed messages between cellphones. These can be in the form of text using SMS (Short Message Service) and can also contain images, video and sound content using MMS (Multimedia Message Service).

As with many other applications, e-communication have some negative implications:

- Messaging has created an easier way for pupils to cheat on exams. Schools and universities have had to put policies in place to stop this.
- Cyber bullying, the spreading of rumours and gossip using various e-communication applications, has become a problem that can cause great distress and damage the reputations of people on the receiving end.

Video Conferencing

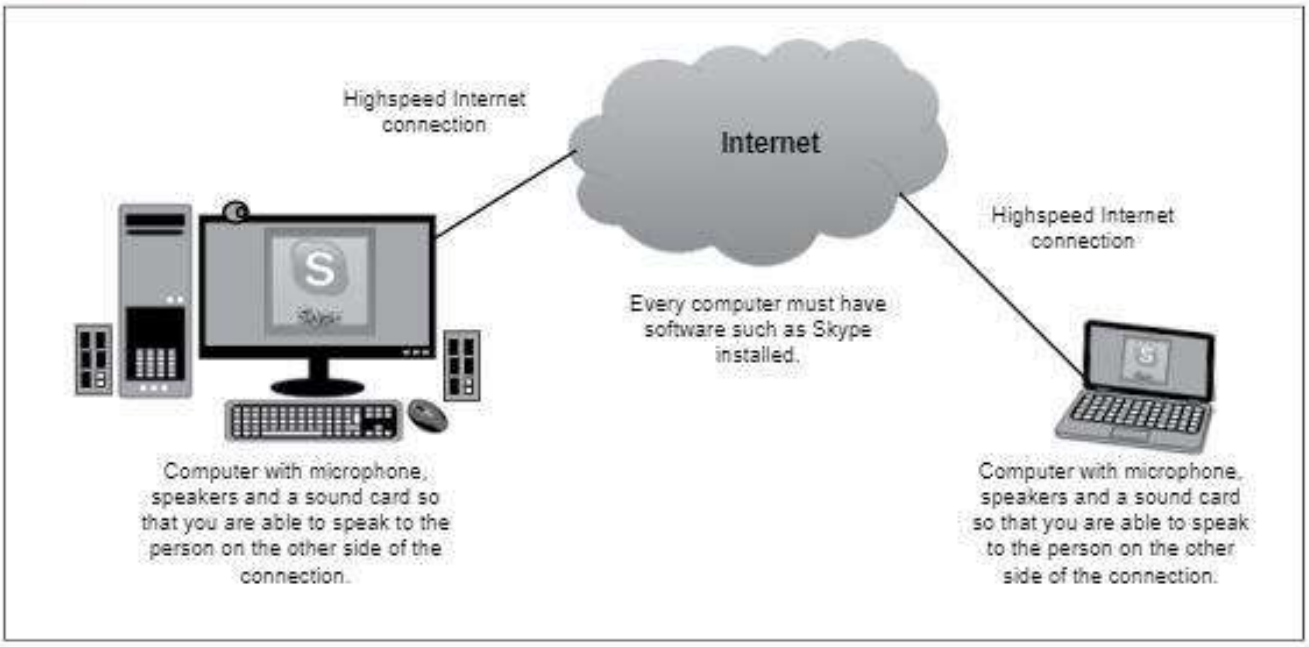
Video conferencing is an application that allows two or more people to hold online conferences at different locations using a computer network to transmit audio and video as if they were all in the same room.

Each participant must have a video camera (webcam), a microphone and speakers linked to their computer. As each person speaks, their voices and images are transferred by the network to all the participant's speakers and monitors respectively.

VoIP (Voice over Internet Protocol)

VoIP is a technology that allows telephone calls to be made over LANs, WANs and the Internet, allowing you to talk to anyone who also has a VoIP system anywhere in the world. When using VoIP, voice is transmitted on a network in the same way in which data is transmitted. It also allows you to speak to many people at the same time using conference call technology.

Skype is one of the most popular software packages to communicate using VoIP. Many people now have their cellphone number, e-mail address as well as Skype address (usually their name) on their business cards. Skype also allows you to make video calls if you have a webcam attached to your computer.



ACtivity • Apply your knowledge

Consider the following e-communication applications:

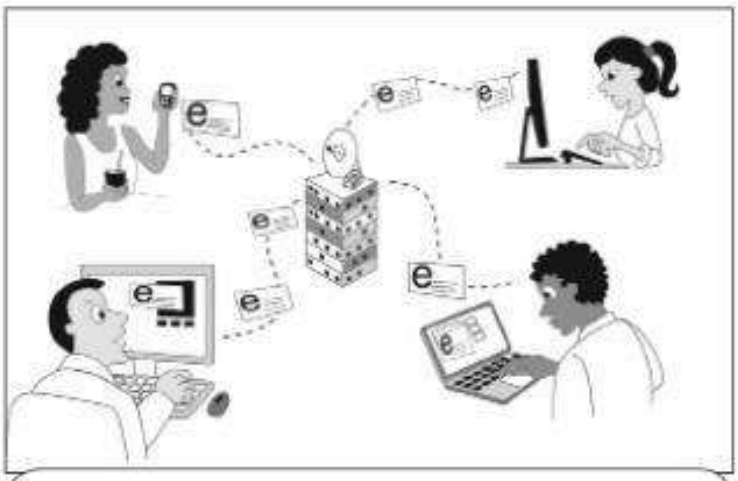
A. E-mail
B. Social networking site
C. IM
D. Skype
E. Video conferencing
F. Online chat

Link each of the following scenarios to the above applications by writing down the appropriate letter:

1. You have family overseas and you would like to be able to talk to them cheaply and be able to see their faces when you speak to them.
2. You have a few friends you would like to keep in contact with, in real-time and with live texting.
3. You are very interested in a topic such as art and would like to communicate with like-minded people and 'talk' to them using a computer in real-time.
4. You are having a birthday party and have designed an invitation. You don't want to print the invitation, but would like to send it to all your friends.
5. A friend has gone overseas on an exchange student program and would like to share his experiences with his friends and family back home. He would also like his friends and family to see photographs of his experiences.
6. Your father's company has decided to cut down on costs and one of the ways is that they no longer want to pay for staff members to travel from all over the country to one venue for a conference.

E-mail

With the growing popularity and availability of the Internet, more and more people are communicating via e-mail rather than by traditional means such as mailing a letter.



An ISP (Internet Service Provider) is a company that has a permanent, fast connection to the Internet and sells Internet access to individuals or organisations for a monthly fee.

Examples of South African ISPs are MWEB, TelkomSA.

E-mail is the composing, sending and receiving of messages electronically over a network including the Internet.

How e-mail works

E-mail works as follows:

- You type the e-mail message on a computer using e-mail software and type in the recipient's address e.g. JoeSmith@icon.co.za.
- Once you send the message it is sent to the e-mail server at your ISP. It is then sent to the e-mail server at the receiver's ISP.
- The message is held at this e-mail server until the recipient checks their e-mail and downloads the e-mail messages.

An e-mail server is used to process, store, send and receive e-mails.

Advantages of e-mail

- E-mail messages can be delivered and replied to much faster (almost immediately) than traditional mail.
- E-mail messages can be sent any time, from anywhere in the world as long as you have an Internet connection.
- It is inexpensive to send and receive e-mail messages.
- It is versatile, as photos and other files can be attached to e-mail messages.
- E-mail protects the environment as it cuts down on the use of paper.

Disadvantages of e-mail

- The recipient needs access to the Internet to receive e-mail.
- Viruses are easily spread via e-mail attachments.
- There is no guarantee the e-mail will be read until the user logs on and checks and downloads their e-mail.
- A user can receive large amounts of spam (electronic junk mail) once his or her e-mail address has been obtained by an institution wanting to advertise.

ISP and web-based accounts

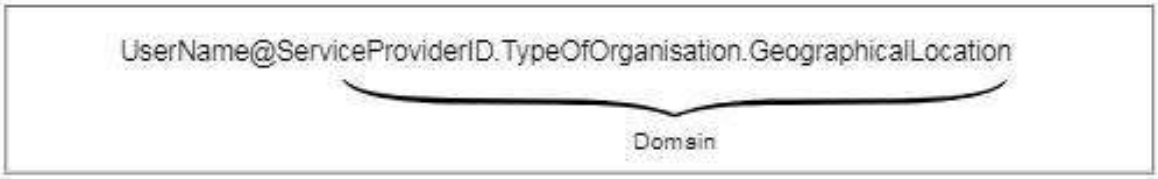
Web-based e-mail or **webmail** is a service that allows you to access an e-mail account through a web browser from any location that has Internet access. You are able to sign up for a free webmail address through providers such as Windows Live Hotmail, Yahoomail (offered by Yahoo), Webmail and Gmail (offered by Google).

When you sign up with an ISP, you will receive e-mail addresses as part of your contract. When using **ISP-based e-mail**, you will access you e-mail by using software such as Microsoft Outlook and Outlook Express on the computer where your account settings have been entered.

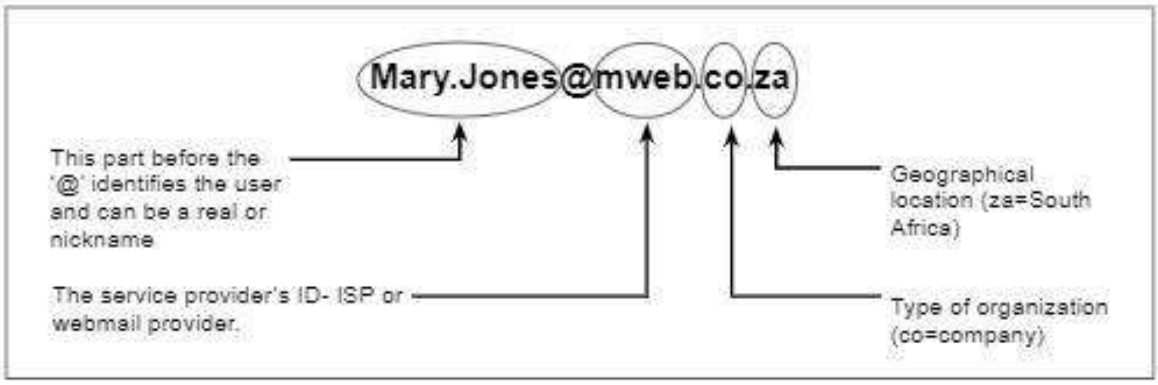
Web-based e-mail	ISP-based e-mail
Web-based e-mail can be accessed using a web browser at any place that has Internet access such as public libraries, Internet cafés, and hotspots in hotels, lodges and airports.	You would usually receive ISP-based e-mail on your own computer using software which has been set up to receive your e-mail. However, most ISPs also offer their clients a web-based service so that they are able to access their e-mail remotely when they are not at their local computer.
You have to stay online to read, create, edit and receive e-mail messages. The messages are stored on the web and are not downloaded to your local computer.	ISP-based e-mail can be accessed without being online as the messages are downloaded to your local computer. You are able to create and edit messages and you only have to go online to send and receive messages.
Webmail is free of charge although you might have to pay for the Internet access.	There is a monthly ISP charge in addition to the cost of Internet access.

E-mail addresses

An e-mail address is made up as follows:



Example:



A few other examples of e-mail addresses are given below:

John.Smith@gmail.com (web-based e-mail address, Gmail is the webmail provider)

John.Smith@telkomsa.net (ISP-based e-mail address, Telkom is the ISP)

Examples of types of organisations

com	Originally intended for commercial organisations and businesses but is now used for any kind of organisation. It is not followed by a geographical abbreviation.
net	Originally intended for sites related to the Internet but is now used for any kind of organisation. It is not followed by a geographical abbreviation.
biz	Businesses of all sizes
edu	Education
gov	Government
org	Non-profit organisation
co	Company
ac	Academic organisations

Examples of geographical abbreviations for country domains

za	South Africa
uk	United Kingdom
sa	Saudi Arabia
au	Australia
fr	France
nz	New Zealand
ca	Canada
us	United States
de	Germany
ls	Lesotho

Practical use of e-mail

For the purposes of this module, we have decided to use Gmail as an example of an e-mail application.

Sending, reading, replying to and forwarding messages

Sending a message

Log into you Gmail account and click on the **Compose Mail** icon.

The following screen will appear:

- The e-mail address of the person to whom you are sending the message goes in the **To** field. If you wish to send the message to more than one person, separate the addresses with a semi-colon or a comma.
- Extra addresses, also separated by semi-colons or commas, can be typed in the **Cc** (**Carbon copy** or **Courtesy copy**) field. All recipients' names are visible to other recipients.
- You are also able to type addresses in the **Bcc** (**Blind cc**) field. A copy of the message will still be sent to these people, but their address will *not* be visible to other recipients in the To and Cc fields.
- It is always good practice to type a short, meaningful description of the subject of the e-mail message in the **Subject** field.
- The message gets typed in the body of the window exactly like typing in a word processor. Remember to always spell check your e-mail before sending it.
- Click on the **Send** icon to send the message. Once the message has been sent, a copy will be stored in the **Sent** folder.

Reading a message

- To read a message go to the **Inbox.**
- Any unread messages will be in bold.
- Double-click on the message to read it.

Replying to a message

Once you have read a message, you may wish to reply to the person.

- With the message open, click on the **Reply** button.
- The **To:** field is automatically filled in with the e-mail address of the person you are replying to. The abbreviation **RE:** is automatically added to the Subject field. This means the message is a reply to the previous message and is about the same subject.
- The **Reply to All** option will reply to all the addresses which are included in the To: and Cc: fields, i.e. to everybody to whom this particular message was originally sent.
- The original message will be visible and you can type in your reply above the message.
- It is good practice to leave the other person's message as part of your reply, as it helps the receiver to keep track of the correspondence, but you can choose to delete it if you wish to.

Forwarding a message

Once you have read a message, you may wish to forward it to your friends.

- With the message open, click on the **Forward** button. The abbreviation **FW** is automatically added to the **Subject:** field. This means the original message with that subject has been forwarded to other users.
- Type in the addresses of the people you wish to send the message to. You can separate the addresses of the recipients with semi-colons or commas.
- It is good practice to delete all the unnecessary addresses of people who may have already received the mail before it is forwarded.

Attachments

Adding attachments

Besides just sending text messages, e-mail applications allow you to attach all sorts of files to the e-mail message. These can include word processing files, spreadsheet files, graphics, video, presentation files, photographs, etc.

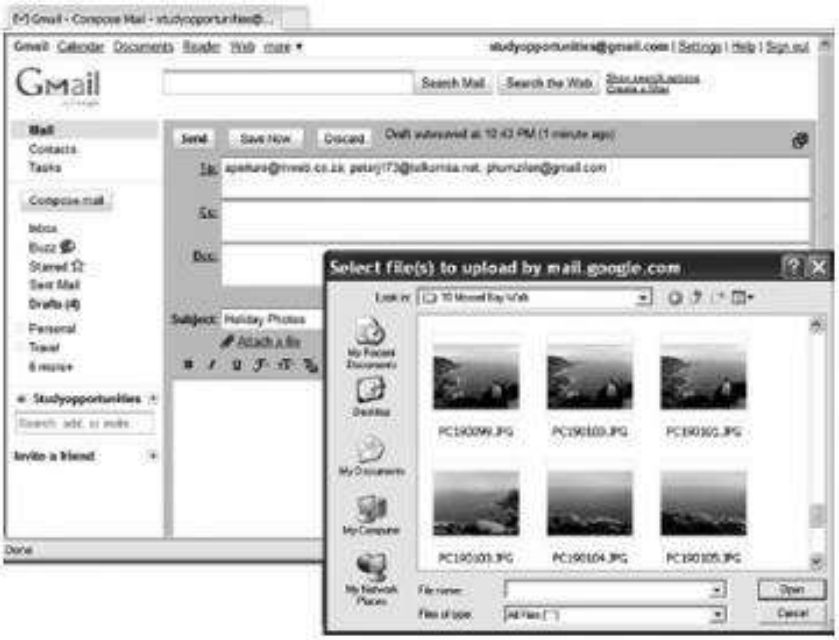
It is good manners to make sure that you do not attach large files to an e-mail message. Should a recipient have a slow connection, it will take a long time for the e-mail to download. Should the recipient have a small cap, downloading many large e-mail attachments could use up their cap quickly, costing them money. The accepted limit for an email attachment is 2 MB or less.

Cap: the amount of data an ISP allows a person to upload and download in a month.

You can use a program such as *WinZip* to compress files before you attach them to an e-mail.

To attach a file do the following:

- Click on the **Attach** icon. It can also have a picture of a paperclip on it.
- A window such as the example will appear:



- You will need to browse for the file on your computer and find where you stored it.
- Once you have found the file (the file name must appear in the *file name window)*, you can attach it – click on **Open**. In some applications you might have to click **Insert** or **Attach** to attach the file, in others it is automatically added. Note that you can attach more than one file at a time to a message.
- Once you have typed your message, click on the **Send** button to send the text message as well as the attached file.

Opening attachments

To open an attachment, open the message from your Inbox.

- The file will be in the attachment window.
- Double-click on the attachment. You can save the file to the hard drive of your computer and then use the correct application to open it. You can also indicate that you want to open the file without saving it first. It will open with the application that it recognises or associates with that particular type of file.

If the contents do not make sense when you open an attachment, or if you cannot open it at all, then you are trying to view a file for which you do not have the correct software installed. You might have to install the software. This is especially true for documents that have been saved with a pdf extension. You will have to install Adobe Reader (it can be downloaded from the Internet) to open this sort of attachment. Adobe Reader documents are documents that can only be read and not edited.

If you receive a message from someone you do not know, be careful of opening attachments as they might contain a virus that could infect your computer. Rather just delete the whole message. Computer viruses will be discussed in more detail in Module 5.2.

2 ACTivity • Test your knowledge

1. Answer true or false:
   a) You can add any sort of attachment to an e-mail message - Word documents, pictures, spreadsheet documents, PowerPoint presentations.
   b) You can only send a message to one person at a time.

2. What is the difference between forwarding a message and replying to a message?

3. Does it matter which one of the Cc or Bcc fields I use to enter the e-mail addresses of additional recipients? Explain your answer by referring to a suitable set of circumstances where you might use each of these options.

4. Does it matter if you use the Reply or Reply to All option when replying to an e-mail?
   Explain your answer by referring to a suitable set of circumstances where you might use each of these options.

5. Your father complains that every time he wants to reply to an e-mail, he first has to delete the original text of the e-mail he is replying to.
   Is it necessary to delete the original text? Briefly motivate your answer.

6. Why do some e-mail messages have the text RE and others FW at the beginning of the subject line?

7. Your friend wants to register an e-mail account and asks for your advice.
   a) Explain the difference between a web-based and an ISP-based account to him.
   b) Name one advantage and one disadvantage of a web-based account.
   c) Give the name of the company that Gmail is associated with.

8. Consider the following e-mail address:
   newspaper@communityhigh.org.za
   a) Explain how e-mail addresses are made up by referring to each of the four sections in the e-mail address above.
   b) Discuss one advantage and one disadvantage of communicating via e-mail.

9. Consider the statement 'E-mail is just so yesterday - IM is the way to go!'
   a) What is IM?
   b) Discuss the above statement by referring to the differences between IM and e-mail.

10. Many schools and companies have banned social networking sites.
    a) Why do you think schools and companies will adopt this policy?
       Try and give a variety of reasons to motivate your answer.
    b) Do you think it is ethical for companies and institutions to do this?
       Motivate your answer
    c) Can you think of any reasons why companies and institutions should not take this action?

Glossary

Cap	Cap is the amount of data an ISP allows a person to upload and download in a month.
Electronic communication (E-communication)	E-communication refers to all forms of communication via electronic means. In other words, it is the process in which computers and other devices such as cellphones are used to exchange messages and/or digital data.
Electronic mail (E-Mail)	E-mail is the composing, sending and receiving messages electronically over a network including the Internet.
Instant Messaging (IM)	IM is an online, text-based communication usually between two people using computers or other devices such as cellphones.
Internet Service Provider (ISP)	An ISP is a company that has a permanent, fast connection to the Internet and sells Internet access to individuals or organisations for a monthly fee.
ISP-based e-mail	When using ISP-based e-mail, you will be able to access your e-mail by using software such as Microsoft Outlook, and Outlook Express on the PC or Laptop where your account settings have been entered.
Messaging	Messaging refers to the exchange of brief typed messages between cellphones. These can be in the form of text using SMS (Short Message Service) and can also contain images, video and sound content using MMS (Multimedia Message Service).
Online chat	Online chat is an application that gives two or more people the ability to have a 'conversation' by typing on a computer using the Internet or cellphone.
Social networking site	A social networking site is where individuals (or organisations) can post their likes, dislikes, interests, photographs and activities for their friends (and virtually everyone else unless access is blocked) to see and read.
Video conferencing	Video conferencing is the ability to hold an on-line conference between two or more people at different locations using a computer network to transmit audio and video data as if they were all in the same room.
Voice over Internet Protocol (VoIP)	VoIP is a technology that allows telephone calls to be made over LANs, WANs and the Internet.
Web-based e-mail (webmail)	Web-based e-mail or webmail is a service that allows you to access an e-mail account through a web browser from any location that has Internet access.

MODULE 3.3

Internet and WWW

The implications of living in the Information Age are enormous. Gone are the days where it was a major task to find relevant, current information. The Internet provides access to a huge amount of information on any subject imaginable. Governments, schools, universities, companies and individuals all offer free information for people to access. Magazines and books are also available on the Internet. We are now immersed in so much information, most of it electronic and freely available, that it has become vital to be able to sift through the vast amounts of information to find exactly what we want. We also need to realise that, although the information might be 'free', we need to acknowledge and give credit to our sources.

Today millions of people around the world use a variety of services or applications that run on the Internet. These include the World Wide Web, e-mail, chat rooms, Voice over Internet Protocol (VoIP) and instant messaging to name just a few.

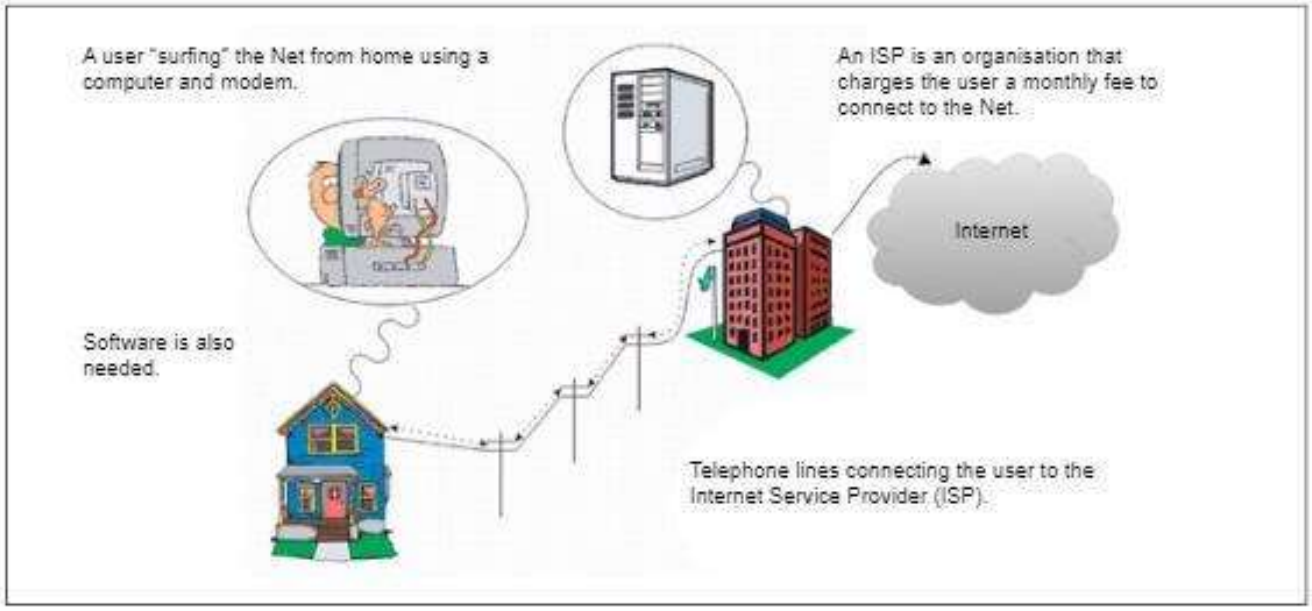
What is the Internet?

The Internet, also called the Net, is a worldwide computer network, consisting of computers and networks that are linked using telephone lines, undersea cables, satellite or microwave for the purpose of communication and sharing of resources. Every computer that connects to the Internet or to a network must have its own unique Internet Protocol (IP) address. An IP address currently consists of 4 sets of digits separated by dots. An example of a typical IP address is the following: 192.168.103.150.

The origin of the term **Internet** comes from the concept of **inter**connected networks.

The Internet and its technologies are changing faster than anyone can keep track of. There is no single organisation in charge of the Net. There are, however, different organisations whose main functions include making recommendations, giving information and registering Internet addresses. The Net is basically kept under control by self-policing, good manners and respect for others.

What is needed to Connect to the Internet?



Computer or other Internet-enabled device

A computer or any other Internet-enabled device such as a cellphone or smartphone is needed.

Internet Service Provider

To connect to the Internet you will need an Internet Service Provider (ISP).

An **ISP** is a company that has a permanent, fast connection to the Internet. These companies sell Internet access and services to individuals or organisations for a monthly fee. This monthly fee is dependent on the type of access that is chosen.

Examples of South African ISPs are MWEB, TelkomSA, MTN and Internet Solutions.

Internet Connection

There are many ways in which an individual or organisation can connect to the Internet. As a computer user you need to know the difference between them to be able to make an informed choice.

Bandwidth or data transfer rate refers to the total amount of data that can be carried from one point to another in a given time period. It is normally measured in bits per second, Kilobits per second (Kbps) or Megabits per second (Mbps).

Broadband refers to any permanent, high speed, high bandwidth connection to the Internet.

Wired connection

ADSL (Asymmetric Digital Subscriber Line) is a permanent digital connection to the Internet using a telephone line. The connection is split into two channels, one for making normal phone calls and one for connecting to the Internet.

An ADSL modem or router is used to connect your computer to an ADSL line that is rented from a telecommunications provider such as Telkom.

Although most modern connection options require you to have a 'modem' of some sort, you need to understand that this is not the same as the modem used to connect using a dial-up telephone line. The term has been kept because most people understand they need a 'modem' to connect electronically to the Internet. In reality these are DTD's (Data Terminating Devices) that use different technologies to send and receive data over different media. Their function is to convert signals from your computer to those transmitted over the telephone line and vice versa so that communication can occur. Today data is transmitted digitally and does not need to be converted to and from analogue signals any more. However, the digital signals used by computers and telephone lines do still differ, and so some measure of conversion is still required.

You have the option of choosing different speed configurations. Below you will see that ADSL connection services have two different speed specifications:

More bandwidth is allowed downstream than upstream. This is because you need a higher speed to download web pages, music or video files than you need to send an e-mail message (you also generally tend to download more than you upload).

ADSL is suited for home use as well as for small and large businesses.

- The costs include a monthly ISP cost, depending on your bandwidth package, as well as a fixed monthly cost for the rental of the line, dependant on the speed configuration that you choose. You do not pay for the connection time to the Internet, but you still pay for normal telephone calls at the normal rate.
- You can access the Internet and make phone calls at the same time.
- You have to check for the availability of the service from Telkom as there is not coverage in every area.
- ISPs offer different data packages which offer different, limited amounts of gigabytes of data to be downloaded and uploaded in a month. This limited amount is referred to as your **cap.** For example, you might have a 3 GB cap using your 384 Kbps line (bandwidth). Uncapped packages are also available.

Wireless connections

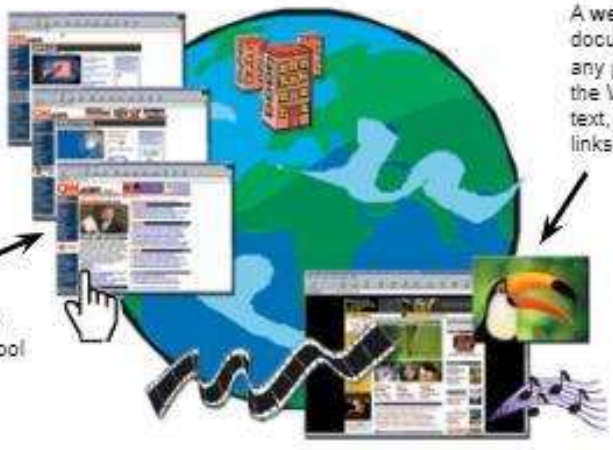
Wireless Internet connections are becoming a popular way of connecting as they offer the freedom of Internet connectivity without having to have an ADSL line installed, as well as allowing mobile access while you are travelling. An example is **3G** which is currently the newest generation of mobile (cellular) phone technologies. A 3G modem or your cellphone is used to connect your computer to the cellular network for Internet access.

The World Wide Web

The World Wide Web (WWW) is a service that runs on the Internet and is a vast collection of interlinked, interactive documents stored on computers all over the world, accessed via the Internet.

The documents are designed in a language called HTML (HyperText Markup Language). This language supports links to other documents, as well as graphics, audio, and video files. These links are called hyperlinks.

Many people think that the WWW and the Internet have the same meaning; however, the World Wide Web (also called the Web or W3) is actually one of the services that runs on the Internet. In the same way e-mail and FTP are also services that run on the Internet.



A **web page** is any electronic document on the Web. It includes any page you access when surfing the Web. Web pages can include text, graphics, sound, video and links to other Web pages.

A **website** consists of related web pages. It is maintained by an organisation such as a school or company.

The first page of a website is called the home page. This home page contains information about the website's purpose and content. It also provides links to other Web pages, websites and documents.

A **hyperlink** is a built-in connection to another related web page and is indicated as text underlined in blue, or with a graphic with a blue outline. To activate a link, move the cursor over the page; when a 'hand' appears, click with your mouse. You are then taken to another page containing additional information and possibly other links. These hyperlinks can be thought of as 'bridges' linking sites on the Web.

After a website has been designed, it needs to be hosted on a web server. A web server is a computer that hosts websites and is connected to the Web. In addition, it may be necessary to employ a person called a webmaster to maintain and develop the website on an ongoing basis.

What is the World Wide Web Consortium (W3C)?

The World Wide Web Consortium (W3C) is an organisation founded and headed by Tim Berners-Lee. Its function is to develop standards so that the WWW reaches its full potential and to ensure the long-term growth of the Web. Its members and fulltime staff work on developing guidelines to work towards this goal.

Uniform Resource Locater (URL)

Every web page has a unique address, called a **Uniform Resource Locator (URL)** or web address.

Remember that an IP address is a numerical address that uniquely identifies a computer either on a network or on the Internet. These can be difficult to remember, so a URL is linked to the IP address to make it easier to remember. You can compare this to having a telephone number listed as 0800-HELP-DESK which in reality links to a telephone number of 0800-4357-3375.

Currently, a web server that hosts one of Yahoo's websites' IP address is 87.248.122.122. You can access this site by either typing 87.248.122.122 or www.yahoo.com in the address bar of a web browser.

A URL is an alpha-numerical address used to identify resources such as web pages, documents and files stored on web servers on the web. The URL is entered in the address bar of a web browser.

In this module we explain how URLs are constructed so that when you come across URLs with various formats, you will be able to interpret them more effectively.

The word **protocol** means a *rule for communication*. There are a variety of network protocols such as the following:

IP	Internet Protocol is the protocol used for communication over a network or the Internet.
HTTP	HyperText Transfer Protocol is the protocol that defines how pages are transferred on the Web. The *http* can be left out when typing in the URL and it will be automatically added in by the browser.
HTTPS	This shows that the site is secure and that financial transactions can be safely carried out on the site.
SMTP	Simple Mail Transfer Protocol is the protocol used for transmission of e-mail over the Internet.
FTP	File Transfer Protocol is the protocol used for the fast, easy transfer of files between remote computers that have an Internet connection. The files can be uploaded to an ftp server or downloaded from an ftp server.
VoIP	Voice over Internet Protocol is the protocol used to manage the delivery of voice information over the Internet. We have already discussed the application of this in Module 3.2.

URLs generally have the following format, however, not all sections are always used depending on the type of URL.

Protocol identifier	://	Type of server	.	Domain name	/	File or resource name
This identifies the protocol to be used to access the resource.	://	*www indicates that the resource is saved on a web server. ftp indicates the resource is saved on an ftp server.*		*This is the unique alpha-numerical address that identifies the website where the resource is located and that is registered to the organisation.*	/	*This substring contains a path to the specific resource or page located on the server.*
http	://	www	.	webopedia.com	/	TERM/W/World_Wide_Web.html
https	://	www	.	fnb.co.za	/	channel/online-banking.html
http	://	www	.	studyopportunities.co.za		*This example has no path as it represents the home page of Study Opportunities*
ftp	://	ftp	.	uar.net	/	pub/e-books/

The domain name can be further explained as follows:

		Domain name				
	.	Site name	.	Type of organization	.	Geographic location
The www indicates that the resource is stored on a web server. Depending on how the URL has been configured, the www can be left out for some addresses and not for others. When in doubt, type in the www.						
www	.	studyopportunities	.	co	.	za
www	.	facebook	.	com		
www	.	cia	.	gov		
www	.	gamesdomain	.	co	.	uk
		wiki.answers	.	com		
		en.wikipedia	.	org		

Types of websites

We are able to classify websites into different categories. There can often be an overlap between the different types, and some websites may fit into more than one category or no category at all. With the ever changing nature of the Internet, the classification of websites may also change.

Portal website

Portal websites, also known as links pages, offer a selection of Internet services from one convenient location. They function as a point of access to other information available on the WWW. This can include a search engine as well as information such as local, national and international news, free hosting of personal websites, weather and sport. Many portals are also Internet Service Providers which offer some of these services on their sites.

Examples include MWEB, Google, MSN and Yahoo.

News website

This type of website contains material relating to current events, news, sport, weather and exchange rates. Many magazines and newspapers maintain websites that contain summaries of articles from the printed version and even articles not found in the printed version.

Examples include CNN, SkyNews, Sunday times, 5FM, 94.7 Highveld Stereo and News24.com

Informational website

Most government institutions have an informational website. These websites contain census and tax information as well as factual information that can be used for research purposes.

Examples of such institutions include The South African Revenue Services and the South African Government.

Business or marketing website

A business or marketing website contains information about selling a product or service.

Examples include: Woolworths, Pick and Pay, Kulula, SAA, Kalahari.net and Telkom

Personal website

Anyone can develop and maintain their own personal website in order to share interests, hobbies and life experiences with the world. Many people are using a blog or a social networking site for their personal website these days.

Weblog (blog)

Weblogs or just plain blogs are websites that are updated often with news and events. A blog reads like a journal with the most recent entry at the top.

Blogs used to be personal but have now become a place for businesses and 'citizen journalists' (people writing reviews, opinions and news, etc. but not employed by a news organization, newspaper, magazine, radio or TV). There are as many different types of blogs as there are people. Some are personal diaries, while others resemble newsletters or columns.

Blogging (or the blogosphere as it is referred to nowadays) allows anyone to publish their thoughts. Some blogs have thousands of readers or subscribers.

The following is a link to a blog maintained by a CAT teacher who shares her teaching and personal experiences online with any interested parties: http://studyopp.blogspot.com

Social networking site

A social networking site is where individuals (or organisations) can post their likes, dislikes, interests, photographs and activities for their friends (and virtually everyone else unless access is blocked) to see and read.

Examples include MySpace, Bebo, Flickr, Facebook, LinkedIn.

Wiki

A wiki is a website that allows people to create and edit web content collaboratively. They can include company intranets, community websites and online reference sites (like an encyclopaedia). A classic example of a wiki is the famous encyclopaedia Wikipedia. The core idea of a wiki is that anyone can contribute to it and that the final published article does not belong to any one individual. Wikis take the form of a web page and access to them is usually free.

Educational website

Educational websites offer ways of formal and informal learning by means of tutorials and simulations. Educators can use the Web as an invaluable source of resource material to make their lessons more interesting. They can also publish assignments and notes for learners to access.

Examples include HowStuffWorks and Webopedia.

Entertainment website

An entertainment website offers all forms of entertainment such as online videos, music, sport, gaming and much more.

Examples include online gaming sites and video sites.

Advocacy website

This type of website describes an idea, opinion or cause and usually presents the views of a particular group or association.

Examples include The S.P.C.A, Cotland's Baby Sanctuary, FreeMe and Greenpeace.

Web application website

A web application is any application that is accessed over the Internet and is not stored on your local computer. Common web application sites are webmail sites such as Gmail, online retail sites such as Kalahari.net and online auctions such as eBay. These applications are very popular as they can be accessed anywhere as long as you have access to the Internet. Google Calendar, a free time-management web application offered by Google, is another example.

Content aggregator

A content aggregator is a website or web application that gathers different web content (such as news, weather, movie reviews and stock prices) from many different online sources.

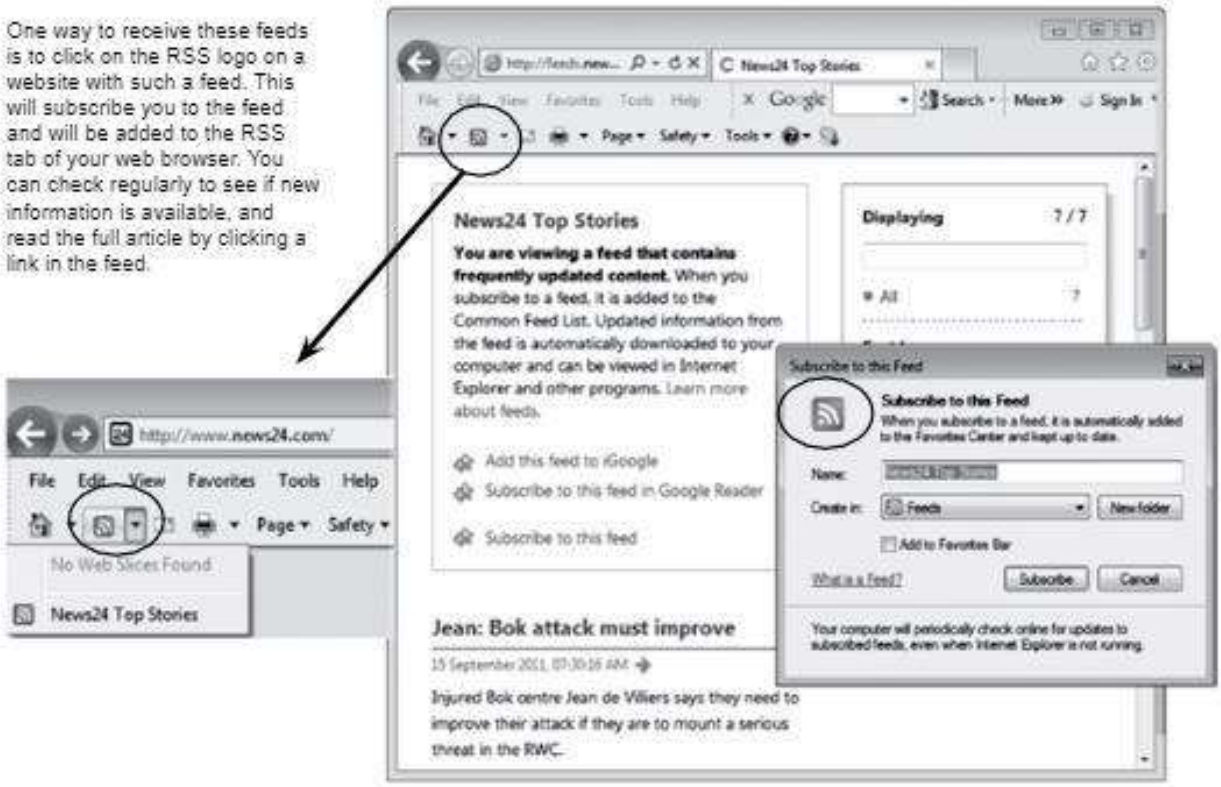
There are two main types of content aggregators.

- The first type is one that gathers information from different sources and displays it on its own website. An example of such a site is the movie review site www.rottentomatoes.com.
- The second type is one that gathers information from various sources and then distributes it. You can subscribe to (become a user of) this type of content aggregator, and choose which information you want to receive. Notification of updates (new information published on the various websites from which the content aggregators receive information) will then be distributed to you (the subscriber) as they become available. (This is called syndication.)

  The content aggregator receives notification of the updates in the form of a document. These documents need to conform to certain formats. This family of possible formats are called RSS 2.0 documents. RSS stands for Real Simple Syndication. RSS documents are called *news feeds*, *channels*, *web feeds* or just *feeds*.

  Web feeds assist publishers (the managers of websites who want their latest information distributed) by being able to update subscribers easily and regularly. Web feeds benefit users by allowing them to subscribe to updates from their favourite websites using one application.

One way to receive these feeds is to click on the RSS logo on a website with such a feed. This will subscribe you to the feed and will be added to the RSS tab of your web browser. You can check regularly to see if new information is available, and read the full article by clicking a link in the feed.



1 Activity

Find examples of at least 5 different websites.

Create a document in your word processor with the following information about each website:

- URL
- Screenshot of home page
- Type of website and motivation

Criteria to evaluate websites

There are many websites found on the Internet which could contain inaccurate or out-dated information. It is therefore important to evaluate websites for quality and relevance before you use the information found on them. There are different criteria to look for and some might be more important for some sites than others. An example of this is the currency criterion. This would be important for a site that has AIDS statistics, but would not be important for a historical site.

Most important is to cross-reference information using more than one site.

The following are examples of criteria that could be used to evaluate websites in terms of content and design.

Authority

- Who is the author and what are his or her credentials?
- Are you able to determine if the author is 'respected' in the particular field of knowledge?
- Is there an institutional affiliation and is it an academic, technical or credible institution?

Accuracy

- Is the information accurate?
- Does the site offer a selected list of sources?

Content / Design

- Is the site easy to navigate and visually pleasing?
- Is the content well organized?
- Is the site user-friendly?

Audience

- Does the site satisfy the needs of its intended audience?

Currency

- Is the information on the web page up-to-date?
- When last was the site updated?
- Are the links kept up-to-date?

Objectivity

- Does the information appear unbiased (Not one-sided, critical of opposing views, etc.)?

Browsing and searching

What is a web browser?

A web browser is a software application that allows one to view and explore web pages on the Web, either via hyperlinks or, in cases where the URL is already known, by typing the address in the address bar of the web browser.

A few examples of modern web browsers include:

Most web browsers have the following features:

- An **address bar** where the address or URL of the web page is typed.
- A **search bar** where you can enter search terms.
- Different **tabs** of the same window or different browser windows are used to open multiple pages or resources at the same time.
- The **Back** button takes you to the page you were previously viewing. If you have viewed many pages, it will take you back one page at a time until you get to your starting point where it will become greyed out and unavailable.

- The **Forward** button will take you forward to the next page if you have viewed other pages.
- The **Stop** button will stop any current operations by the browser. Any file that is currently loading will stop loading. It can also be used to stop animations from continuing once the page has loaded.

Web pages that you have accessed before are often stored temporarily on your hard drive by the operating system to save time by not having to fetch them from the Internet again. For this reason your browser might be showing you an 'out-dated' version of the web page, which is then updated when you press the Refresh or Reload button. The function key is F5.

- The **Refresh or Reload** button will reload the current web page that is being viewed. It is useful especially for pages that you view often, as this button will allow you to view any changes to the page as soon as they are available. The Refresh button can also be used to reload a page that may have been interrupted.

- The **Home button** will return you to the page that you have set up as your default or home page, in other words the first page that appears when you load your browser.
- The **Favorites** or Bookmark menu item allows you to store shortcuts to web pages.
- The **History** menu item allows you to find web pages you have previously visited. When you select the item, it brings up a list of the most recent sites visited. You can configure how long you want to keep the history. The history is also shown in the drop down menu of the address bar.
- The **Print** menu option allows you to print or print preview the current page that you are viewing.

What is a search engine?

A search engine is a program that is used to search for documents located on the WWW by using keywords or phrases entered by the user when looking for information.

Search engines use programs called 'spiders' or 'robots' to 'crawl' the web and log keywords from web pages for which searchable databases are then compiled.

When using a search engine, the user types a keyword into a search box. The database is then scanned and all the websites in the database containing that word or phrase are returned. We call this a 'hit'. Different search engines use different methods or algorithms to look for 'hits'. This, coupled with the fact that each search engine has different pages 'catalogued', means that you will get different 'results' from different search engines. The 'hits' are generally returned in order of relevance to the query, although companies can pay to have their pages ranked higher!

The following are a few examples of search engines:

Google South Africa www.google.com	YAHOO! www.yahoo.com	Ask www.ask.com	bing www.bing.com

Tips for successful searching

The following steps can be followed when using a search engine to look for information on the WWW.

Suppose you are looking for information on Beluga whales for a Life Sciences project.

Step 1: State exactly what you want to find.

For example: What characteristics do Beluga whales have?

Step 2: Identify keywords.

It is important to identify keywords in order for you to refine your search. If you were using the Google search engine and typed in just the word *Beluga*, you would get about 6 million hits, as 'Beluga' refers to caviar, whales, and could also include names of restaurants and schools for example.

If you typed in the keywords *Beluga whales*, you would get about 800 000 hits, which has already refined your search a great deal.

If, however, you look for the underlined keywords in the following sentence:

What characteristics do Beluga whales have, the number of hits in your search drops to about 46 000. Most search engines ignore 'common words' such as 'the' and 'a'.

From this example you should therefore notice that the more *relevant* keywords you type in, the more refined your search will become.

Step 3: Use synonyms

You may need to use synonyms. (A synonym is a word that means the same or nearly the same as another word.)

Examples of synonyms in the above example: Characteristics: features, qualities.

Step 4: Check your spelling

Search engines return websites with words that match the keywords that you type in. Whether you type a word in lowercase or uppercase usually does not matter. Make sure that you spell the words correctly. Some search engines, such as Google will try to return the correct spelling if it can recognise the word from your spelling.

Step 5: Use advanced search options

Most search engines have advanced search facilities where you can specify amongst other things:

- The date when the page was created/changed (useful if you are looking for up-to-date information).
- The type of file you want to look for e.g. you may want to find a spreadsheet with data on HIV statistics.
- The domain you want to search e.g. only in the South African (ZA) domain.
- Using operators such as 'or' to refine your search.

Give credit where it is due

It is important to be aware of issues involving copyright, intellectual property and plagiarism whilst conducting research. It is not only important to supply a record of your sources but also to be aware of what you may and may not use as part of your research. Plagiarism is committed when someone else's words or ideas are used and the source is not acknowledged.

Acknowledging a website source

Author's surname, name (if known). *Title of website*. Date on which the website was posted or updated. [URL] Date on which the website was visited by you.

Example

Meevis, Hans. *Jewellery Gallery*.April 2011. [http://www.meevis.com/jewelry-gallery-0601.htm] 21/05/2011

Searching for images

When using a search engine such as Google, type in the keyword and click on the images link to find relevant images for your topic. You can also specify the size and type of graphics file you are looking for.

Right-click on the picture and a drop down menu will appear.

- To save a picture, choose **Save Picture As** and choose the location you wish to save the picture to.
- You can also *copy* the picture and *paste* it in an application such as *Word*.

2 Activity

The following is a list of computer terminology:

Phishing, Spam, Cookies, Firewall, Internet Café, Spyware, Pop-Ups, Flash, QuickTime, ActiveX

Find a short explanation on the Internet about three of the above topics and do the following:

- Create a folder with a suitable name for this activity.
- Save the web pages into your folder.
- Find an image for each topic and save the images in your folder.
- Copy the information to a *Word* document and save the document in the folder.
- Insert the images into the relevant sections in your *Word* document.
- While you are working create a bibliography of the sources you use and add it to the end of your document.

3 Activity: Understand and apply

1. The Internet is a worldwide computer network. How is each computer that is connected to this network identified?
2. Briefly explain what an ISP is and give the name of a South African ISP. (Do not simply expand the abbreviation in your answer.)
3. ADSL
   a) Describe clearly what a broadband connection is.
   b) To what do the following ADSL specifications refer: 512 Kbps / 128 Kbps?
   c) What is a cap?
   d) Discuss two advantages of ADSL.
   e) What does the monthly cost for ADSL include?
4. Describe clearly what the function of a Data Terminating Device ('modem') is.
5. Wireless Internet connections
   a) To what does 3G refer?
   b) Discuss two disadvantages of 3G over ADSL.
6. Clearly explain the difference between the WWW and the Internet.
7. Distinguish between a website, web page and home page.

8. Some URLs begin with the prefix ftp:// instead of HTTP.
   What is the significance of the ftp?

9. While looking for information on a project you went onto the following website: http://www.customs.gov.uk. You click on the 'contact' hyperlink to see how to contact someone on the site.
   a) Would they use a search engine or a web browser to go to this website? Motivate your answer by explaining the difference in function between these two.
   b) What can you deduce from the *gov.uk* part of the web address?
   c) What does the *www* mean in the address? (Do not simply expand the abbreviation in your answer.)
   d) Explain briefly what a hyperlink is and how it can be found or seen on a web page.
   e) Give a suitable example of an e-mail address for the domain or website above.

10. Describe one way in which you would be able to transfer a large file to a computer across the world without attaching it to an e-mail.

11. Explain how you can receive an RSS feed.

12. "Don't believe everything you read on the Internet."
    State three guidelines to follow in an effort to verify the validity of information on the Web.

13. How does a URL differ from an IP address? Give an example of each to clearly show the difference.

14. Give two visual indications to look out for, visible on the web browser, to satisfy yourself that the site is indeed secure.

15. A friend complains that she knows the school's web page has been updated but she cannot see the changes. Explain what the possible problem is and how she can rectify this problem.

16. The Back button is 'greyed out. What does this mean?

17. I often go to a particular website which is very slow to access as there is a very fancy animation on the home page. Is there any way I can access this page any quicker?

18. I want my browser to automatically go to a particular search engine when I open my web browser application. How can this be done?

19. 1I came across a really interesting website two weeks ago, but I forgot to write down the name of the website. It there any way I can find out the name of this website?

20. Your younger brother is doing a project for Arts and Culture about the Ndebele beadwork. He copied and pasted some pages including some pictures from the Internet into a Word document and added a cover page with his name and title of the project. Explain to him in simple terms why he is not allowed to copy information in this way.

Glossary

3G	3G is currently the newest generation of mobile (cellular) phone technologies.
Asymmetric Digital Subscriber Line (ADSL)	ADSL is a permanent digital connection to the Internet using a telephone line.
Bandwidth	Bandwidth or data transfer rate, refers to the total amount of data that can be carried from one point to another in a given time period.
Broadband	Broadband refers to any permanent, high speed, high bandwidth connection to the Internet.

Cap	Cap means that you are limited to a certain amount of gigabytes of data downloaded and uploaded on a month.
Data Terminating Device ('Modem')	The function of a 'modem' is to convert signals from your computer to those transmitted over the telephone line and vice versa so that communication can occur
File Transfer Protocol (FTP)	File Transfer Protocol (ftp) is the protocol used for the fast, easy transfer of files between remote computers that have an Internet connection.
FTP server	A computer allowing people to use FTP to upload or download files.
Home page	A home page is the first page of a website and contains information about the website's purpose and content. It also provides links to other web pages, websites and documents.
Hyperlink	A hyperlink is a built-in connection to another related web page, and is indicated as text underlined in blue, or by a graphic with a blue outline.
HyperText Transfer Protocol (HTTP)	HyperText Transfer Protocol (http) is the protocol that defines how pages are transferred on the web.
Internet	The Internet, also called the Net, is a worldwide computer network, consisting of computers and networks that are linked using telephone lines, undersea cables, satellite or microwave for the purpose of sharing resources and communication..
Internet Protocol (IP)	IP (Internet Protocol) is the protocol used for communication over a network or the Internet
Internet Protocol Address (IP address)	Every computer that connects to the Internet or in a network must have its own unique Internet Protocol (IP) address.
Internet Service Provider (ISP)	An ISP is a company that has a permanent, fast connection to the Internet. They sell Internet access and services to individuals or organisations for a monthly fee.
Search engine	A search engine is a program that is used to search for documents located on the Internet by using keywords or phrases entered by the user when looking for information.
Uniform Resource Locator (URL)	A URL is the unique address of a web page.
Web browser	A web browser is a software application that allows one to view and explore web pages on the Web either via hyperlinks or in cases where the URL is already known, by typing the address in the address bar of the web browser.
Web page	A web page is any electronic document on the Web. Web pages can include text, graphics, sound, video and links to other Web pages.
Web server	A web server is a computer that hosts websites and is connected to the web.
Website	A website consists of related web pages.
World Wide Web (WWW)	The World Wide Web is a service that runs on the Internet and is a vast collection of interlinked, interactive documents stored on computers all over the world, accessed via the Internet.
World Wide Web Consortium	The World Wide Web Consortium (W3C) is an organisation founded and headed by Tim Berners-Lee. Its function is to develop standards so that the WWW reaches its full potential and to ensure the long-term growth of the Web.

MODULE 3.4

Web page technology

Introduction

You may spend a lot of time on the Internet – browsing the Web, chatting with friends, facebooking, googling, keeping up to date with all the topics that interest you. If your parents trust you, then you may have even bought something over the Internet as well!

You use this technology – but do you have any idea of how it works? We are going to use this module to take a look at HTML, the way that web pages are formatted. You will write code yourself to

- format text on a web page
- display pictures on a web page
- link to other web pages.

What is a web page?

A web page is a text file. It's as short and simple as that.

1 Activity

Try this if you don't believe that a text file can be a web page:

- Go to a text editor (even Microsoft Word).
- Create a new document.
- Type in 'Hello world - this is my first web page!'
- Save the document as a text file and close it.
- Rename the file (in Windows Explorer) so that you change its extension from .txt to .html.
- Double click the file and it will open in your browser and you will see the text you typed.

Web pages are text files for a reason – it guarantees maximum compatibility between all the different types of computers, devices and operating systems that need to be able to access and display them.

The problem is that plain, unformatted text can look very boring and uninteresting. There had to be a way to

- add some type of formatting to the text
- provide access to media files (pictures, sounds, etc)
- add links to other documents into the page.

The problem is that all this had to be done whilst still using text files only. So a format was defined along with rules of how a text file would be 'understood' and displayed by the browser software that would open and read the web pages. That format is HTML (Hypertext Markup Language).

HTML is not the only formatting used in modern web pages.

Other formats used include:

XML: Extensible Markup Language adds a way for web pages to manage data dynamically.

CSS: Cascading Style Sheets are a way of creating standard formatting that can be applied to all the web pages on your site, and so reduce the amount of formatting and code you have to put into each individual page.

Javascript: Is an example of a scripting language. Scripting languages are normally interpreted by another program at runtime. JavaScript may be embedded within HTML to provide some logical functionality, to the web page. People often confuse JavaScript and the Java programming language, they are in essence two completely different programming languages, with some resemblance in syntax. Examples of other scripting languages are, ASP (Active Server Pages) and PHP (Hypertext Preprocessor).

AJAX: A technology combining XML and Javascript to make web pages even more like native computer programs.

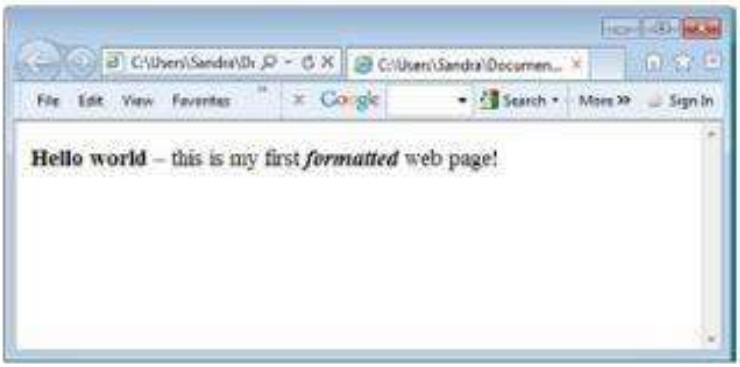
HTML

HTML is nothing more complex than a set of formatting codes that you can put into a text file that will change the way the browser displays your text.

2 Activity

- Use a text editor or a word processor to create a new document.
- Type the following:
  Hello world - this is my first formatted web page!
- Save the document as a text file and close it.
- Rename the file so that you change its extension from .txt to .html.
- Double click the file and it will open in your browser and you will see the text you typed.

In Activity 2 you have added HTML formatting ***tags*** to your web page. What you should see in your web browser is something that looks like this:



What you have saved is all text, but all the stuff inside the angled brackets (the < and > signs) is regarded as formatting by the web browser and changes the appearance of the text when the browser displays it.

Hello world – this is my first formatted web page!

HTML 'programming' is the art of formatting text files so that they display as meaningful, attractive, well-designed, user-friendly websites. The range of tags that you can use is large and using them properly can be quite tricky – so there is an art to creating a good web page.

Basic formatting tags

Let's just take a quick look at the most simple and basic formatting tags:

	Start of formatting	End of formatting	Comment
Basic document tags	<html>	</html>	Identifies the document as an HTML document. Part of formal structure. Does not have to be included.
	<head>	</head>	Marks the text as a header.
	<title>	</title>	Marks the text as a title – this appears in the title bar of the browser and in the tab of a tabbed browser.
	<body>	</body>	Marks the text as body text in the document.
Heading elements	<h1> <h2> <h3>	</h1> </h2> </h3>	These create headings that are displayed using the browsers default formatting for heading level 1, heading level 2 and heading level 3.
Text elements	<font size="+1" color="#ff0000">		An example of enlarging a font and changing its colour (1 size larger than normal and the colour is red). The colour format is #rrggbb where the two digits rr represent the red value in hexadecimal, the gg, green and the bb = blue.
			Marks the start and end of a paragraph. The is not essential. Starts a new paragraph even if the was not used.
	 		Adds a line break without starting a new paragraph.
			Marks text as bold.
			Marks text as italic.
Text alignment	< align="left">		This is the default if no alignment specified. Aligns text left.
	<align="center">		Centers text.
	< align="right">		Aligns text right
Image			Inserts a picture into the page where the name of the file comes after **src** and the text to be displayed if the image cannot load, comes after **alt**.
Ordinary links	Link-text 		A link to another web page – **Link-text** should be replaced by the text to appear as the link to the web page. The example shows a link to a web page on the internet. If you just want to link to another html document in the same folder that your current page has loaded from, you should use the name of the file – as you can see in this example: Link-text
Bullets	 - Keyboard - Mouse - Microphone 		Creates a bulleted list where the whole list starts and ends with and individual list items start with and end with .
Numbering	 - Keyboard - Mouse - Microphone 		To make the list numbered use and instead of the <u;> used for bullets.

Syntax and order is important!!!

You may nest (put tags inside each other) tags to combine them, but the end tags must be placed in **exactly the reverse order**.

For example: to make text bold and italic you would use this text has two formats .

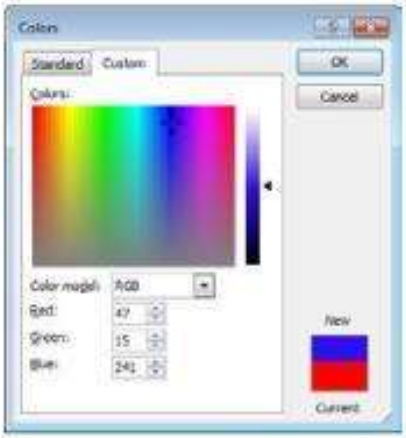
If you used this order: this text has two formats you have ended the tag sequence incorrectly, and this will produce unexpected results in the way your text is displayed.

Some text and symbols have meaning as tags and you can't type them as text – instead, you need to use special tags to display them. Here are a few for you to look at:

Code	Symbol
&lt;	<
&gt;	>
&amp;	&
&quot;	"
	space
&pound;	£
&euro;	€

Tip:

You can use the *Colour Picker Tool* of *Word* to choose the colour you would like to use. Convert the values for red, blue and green to hexadecimal.



3 Activity

Open the file **BusiCoffeeMainIndex.htm** with a web browser. This is an example of the type of web page you need to be able to create.

Look at the source code (HTML instructions) of the website. You can do this by opening the htm file in NotePad, or you could view it in the browser itself.

Browsers use different options to display HTML code:

- In Internet Explorer, you would view it by clicking **View, Source.**
- In Mozilla FireFox, you would click on the FireFox tab, then choose **Web Developer, View Page Source.**

Characteristics of a well designed web page

Designing and planning a good web page has a lot in common with planning and designing a good user interface for your program. Remember that the readers of a website want a website that

- is easy to understand
- is easy to control or input data
- makes it easy to see the results
- is logical in flow and structure
- gives clear instruction or cues about where you can do what
- is simple – simple is always better than complicated.

This is a whole area of study in itself and combines practical usability with an artistic sense of what looks pleasing and attractive for its target audience. Here we give you a few aspects to consider:

Readability

Readability is more important than having a fancy web page with lots of bells and whistles. People will get irritated with content that is hard to read, and simply stop looking at your site rather than think how wonderfully talented and creative you are!

Don't do THINGS like use lots of different fonts an colours simply, **because you can**!

TEXT ALL IN CAPITALS IS CONSIDERED RUDE, AS IT IS REGARDED AS SHOUTING AND IS HARD TO READ.

Centred text may be good for headings but
paragraphs of centred text with lots of lines underneath each other
with lots of things to read does not work very well
at all!

The following can be done to improve readability:

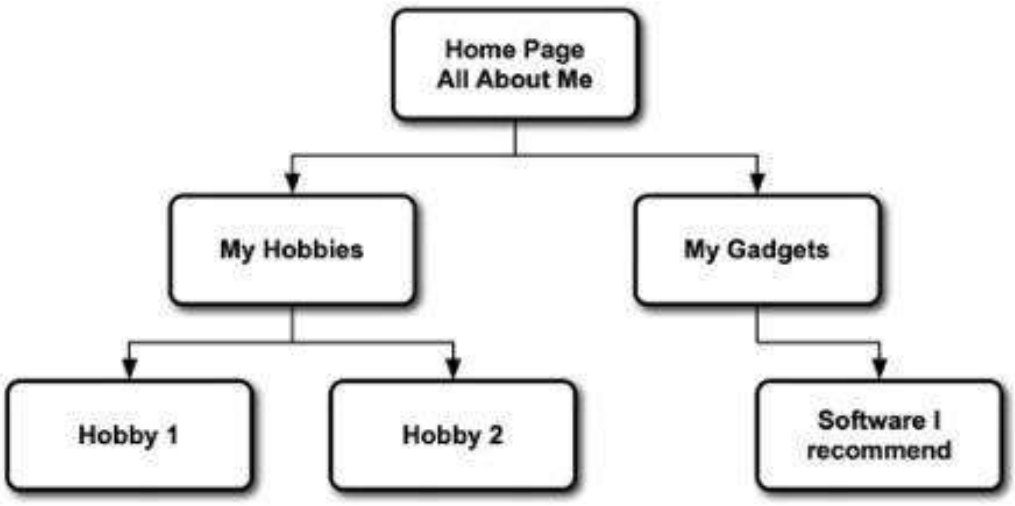
- Use paragraphs to break up long text.
- Use numbered or bulleted lists to make content more accessible.
- Group related ideas and content in logical order.
- Clearly separate unrelated items.
- Use emphasis sparingly – only when it makes a valid point.

Navigation

This means that users should find it easy to move around on your website. They should be able to get to all the pages easily and get back to where they came from just as easily. The links should be clear, well positioned and users should be able to understand where they lead to. To do this you need to

- plan your website
- think of your site like an organisational diagram, with the Home Page at the top and the others arranged around it (see the example)
- make sure that there are links on every page in the website that allow you to navigate the site – or at least get back to the main page.

Planning a website



Consistency

Don't use different layouts and fonts and colours on different sections of the page simply because you can. Use them if they serve a purpose. Consistency of design and layout makes it easier for the user to understand what is going on. Your site should also have consistency between all the different pages on the site. That includes at minimum:

- Using the same basic layout (headers in the same place, navigation in the same place, etc.)
- Using the same colours and fonts
- Using the same order of items on the page

Layout

Layout should try to group related elements together. It should flow logically and make sense to the user.

Why not, for example,

- put all navigation links in a bar at the top of the page just under the heading?
- for long documents, put links to separate parts in a separate table down the left side?
- use tables to merge your text and images? (Tables are a little more complicated and if you want to use them then you should investigate how they work. Tables are great tools for organising the layout of a page.)

Typography

Typography involves the placing and formatting of text. This includes aspects like the length of a line of text, the type and size of the font, the amount of space between letters, words and lines, and the use of punctuation. On a very basic level, you need to make sure that your font types

- are easy to read.
- match the content that appears in them – you can't do a serious message in the Comic Sans font, for example.
- are fonts that are likely to be on the user's system or they will end up seeing a font other than what you have chosen.
- are limited in number – too many fonts on a page are difficult to read and break the rule of consistency.

4 Activity • Create a basic web page

Using the formatting tags shown above, use your favourite text editor to create

1. a web page introducing yourself
2. a web page describing your three main hobbies or interests

Make sure that the web pages contain links to each other.

5 Activity • Apply your knowledge

1. Indicate whether the following statements are true or false. If the answer is false, re-write the statement so it is true.
   a) HTML is an abbreviation for: Hypertext Manipulation Language.
   b) The tags in HTML are not case sensitive.
   c) In order to display an & as part of your web text, the special tag & should be used instead.
   d) Using the <ol> and <li> will create a list, with bulleted items.
   e) The following tag is valid in HTML: <font size="+1" color="#ff0000">

2. Identify and correct the errors in the following HTML code which aims to render the page as displayed below:

Code with errors:

```
<HTML>

<HEAD>
<TITLE>My Info Page Page</TITLE>

<BODY>
<font  colour="#ff0000>
<h1>  Hi there! my name is Lerato Madibeng.</h1>

<font colour="#0033CC">
<p> I live with my granddad and grandma in: <b> Giyani. </b></p>

<p1>My three favourite subjects at school are:</p1>
<ul>
<li>Math</li>
<li>Science</li>
<li>Information Technology</li>
</ul>

<p1>My brother is a systems developer, he writes computer applications in Delphi</p1>

</HTML>
```

3. Study the web page.
   a) Write down 5 good design principles that were applied by the developer of the site.
   b) List 2 ideas which could be implemented to make the website better.

Glossary

HTML (Hypertext Markup Language)	A set of formatting codes, placed in a text file to change the way a browser displays text.
Tags	Formatting codes placed between '<' and '>' in a text file.
Text file	A file that contains ASCII text only and no formatting.
Typography	Typography involves the placing and formatting of text. The choice and use of fonts is an important aspect of typography.

Section 4

Data and Information Management

Modules

Module 4.1 Data as the computer sees it
Module 4.2 File management

MODULE 4.1

Data as the Computer sees it

When using computers, we do not have to worry too much about what happens 'deep inside' the computer. However, one needs to understand some of these concepts as they do have implications and consequences for us at a higher level, especially in terms of the accuracy of data.

Number systems

We are familiar with and used to using the decimal or base 10 number system. There are, however, also some other number systems that are important to computers. Because of their electronics, computers work with only two states – on or off, that is a binary or base 2 number system. We shall be looking at some of these other number systems in this module. Fortunately they are all based on much the same principles.

Number systems most used in the Computer environment

Number systems using other bases, such as base 2 (binary), base 8 (octal) and base 16 (hexadecimal) are often used in the computer environment. We are going to look how the binary and hexadecimal systems compare with the decimal system.

Base 10 (Decimal)	Base 2 (Binary)	Base 16 (Hexadecimal)
Each number system uses different digits and/or symbols. These digits start at 0 .. up to (base–1). The following digits/symbols are used by the various number systems:		
0,1,2,3,4,5,6,7,8,9	0, 1	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F Because the numbers 10 to 15 consist of more than one digit, the symbols A, to F are used as digits to represent these numbers.
Numbers are written with the base to the right in subscript (with the exception of base 10). Examples of numbers from each number system are:		
2358	1010101_2	$A2CD3E._{16}$

All numbers use the concept of 'place value', where the value of each digit in the number depends on its position in the number.

Let's take the number 111.11 for example:

1	1	1	.	1	1

Each of the 1's represent a different value depending on its position within the number. E.g. the 1 on the extreme left represents 100 and the first 1 to the right of the decimal point represents 1/10 or 0.1

The following examples illustrate the place values for each digit in a number. Note how the expanded notation of a number gives one the decimal value of the value. (The placeholder column number indicates the position of the digit in the number.)

Decimal number (base 10): 4192.304

Example of number	4	1	9	2	.	3	0	4
Placeholder column	3	2	1	0		-1	-2	-3
Place value written as Base $^{\text{Placeholder Column}}$	10^3	10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}
Place value	1 000	100	10	1		1/10 =0.1	1/100 =0.01	1/1000 =0.001

Expanded notation: 4192.304

$= 4x\ 10^3 + 1x\ 10^2 + 9\ x10^1 + 2\ x\ 10^0 + 3x\ 10^{-1} + 0\ x\ 10^{-2} + 4\ x\ 10^{-3}$

$= 4000 + 100 + 90 + 2 + 0.3 + 0 + 0.004$

$= 4192.304$

Binary number (base 2): 1101.101

Example of number	1	1	0	1	.	1	0	1
Placeholder column	3	2	1	0		-1	-2	-3
Place value written as Base $^{\text{Placeholder column}}$	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
Place value (as a decimal value)	8	4	2	1		½ =0.5	¼ =0.25	⅛ =0.125

Expanded notation: 1101.101_2

$= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

$= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times 0.5 + 0 \times 0.25 + 1 \times 0.125$

$= 13.625$

Hexadecimal numbers (base 16)

Example of number	1	A	5	F	.	3	D
Placeholder column	3	2	1	0		-1	-2
Place value written as Base $^{\text{Placeholder column}}$	16^3	16^2	16^1	16^0		16^{-1}	16^{-2}
Place value (as a decimal value)	4096	256	16	1		1/16 =0.0625	1/256 =0.00390625

Expanded notation $1A5F.3D_{16}$

$= \mathbf{1} \times 16^3 + \mathbf{A} \times 16^2 + \mathbf{5} \times 16^1 + \mathbf{F} \times 16^0 + \mathbf{3} \times 16^{-1} + \mathbf{D} \times 16^{-2}$

$= \mathbf{1} \times 16^3 + \mathbf{10} \times 16^2 + \mathbf{5} \times 16^1 + \mathbf{15} \times 16^0 + \mathbf{3} \times 16^{-1} + \mathbf{13} \times 16^{-2}$

$= 4096 + 2560 + 80 + 15 + 0.1875 + 0.05078125$

$= 6751.23828125$

The importance of binary numbers

As computer hardware was developed, engineers discovered that it was easy, from a 'physical', engineering point of view, to have just two states – on or off. This could easily be represented by the presence or absence of current flow. Hence, at the lowest level, data is represented in binary, to make it easier to design and build hardware.

Converting from another number system to the decimal system

Convert binary numbers to decimal numbers

To convert a binary number to a decimal number, the following method can be applied:

1001_2 $= (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$ (Write the number in expanded notation.)
$= (1 \times 8) + (0 \times 4) + (0 \times 2) + (1 \times 1)$ (Calculate the decimal value.)
$= 8 + 1$
$= 9$

1 ACtivity • Apply your knowledge

Convert the following binary numbers to decimal numbers, showing all your calculations.

1. 1011_2 2. 111101_2 3. 1000100_2 4. 10111001_2 5. 11001110_2 6. 100111100_2

Convert hexadecimal numbers to decimal numbers

$2F3_{16}$ $= (2 \times 16^2) + (F \times 16^1) + (3 \times 16^0)$ (Write the number in expanded notation.)
$= (2 \times 256) + (15 \times 16) + (3 \times 1)$ (Calculate the decimal value.)
$= 512 + 240 + 3$
$= 755$

Hint: You can check your answers using a scientific calculator such as the one supplied in Windows (under Accessories).

You will notice that the smaller the base, the longer a number tends to be (contains more digits), which is why computer programmers often use hexadecimal instead of binary, as it is quicker to write down $A1CF5C_{16}$ than $10100001110011110101110 0_2$.

2 ACtivity • Apply your knowledge

Convert the following hexadecimal numbers to decimal, showing all your calculations.

1. $A7_{16}$ 2. $D15_{16}$ 3. 100_{16} 4. AAB_{16} 5. $34E_{16}$ 6. $5E8_{16}$

Converting a decimal number to another number system

We have seen how we can 'interpret' binary and hexadecimal numbers to obtain their decimal values. Now we are going to do the opposite and convert decimal values to binary and hexadecimal values.

Convert decimal numbers to binary numbers

To convert a decimal number to a binary number, you can use one of the following two methods: (We are going to demonstrate the conversion using the decimal number 634.)

Method 1

Create a table representing the place values of the binary system. (You need to include only the place values smaller than 634.)

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1

Now, starting from left to right, find values that add up to the total desired, in this case 634. In the bottom row, you place a 1 for every number used and a 0 if it is not used.

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
512	256	128	64	32	16	8	4	2	1
1	0	0	1	1	1	1	0	1	0

The numbers you chose should add up to 634 = 512 + 64 + 32 + 16 + 8 + 2.

Therefore the binary equivalent to 634, is 1001111010_2

Method 2

In this method, continuously divide the number (634) by the base number (2 in this case). Write down the remainder of each division.

634	÷ 2 = 317	remainder 0
317	÷ 2 = 158	remainder 1
158	÷ 2 = 79	remainder 0
79	÷ 2 = 39	remainder 1
39	÷ 2 = 19	remainder 1
19	÷ 2 = 9	remainder 1
9	÷ 2 = 4	remainder 1
4	÷ 2 = 2	remainder 0
2	÷ 2 = 1	remainder 0
1	÷ 2 = 0	remainder 1

Please note: The answer is obtained by writing down the remainder of each division, from the last to the first.

The answer is 1001111010_2

3 Activity • Apply your knowledge

1. Convert the following decimal numbers to binary numbers.

   a) 25 b) 67 c) 99 d) 126 e) 139 f) 199

2. Write your birth date using binary numbers. Obtain the binary version of the birth date of the person sitting next to you and convert it to the regular notation (decimal):
   e.g. 1997/06/12 = 11111001101/110/1100

Convert decimal numbers to hexadecimal

To convert a decimal number to a hexadecimal number, you can use one of the following two methods: (We will again demonstrate the conversion using the decimal number 634.)

Method 1

Create a table representing the different place values of the hexadecimal system. (You need to include only the place values smaller than 634.)

16^3	16^2	16^1	16^0
4096	256	16	1

Similar to binary, you have to find numbers adding up to the total desired. However, since hexadecimal include values from 0 to 15 and not just 0 and 1 like binary, you need to determine the number of groups of the place values required.

Find the first place value less than your start value (in this case 256 is the first place value less than 634). Determine how many groups of the place value can 'fit into' the value to be converted.

256 x 3 = 768, so 3 groups are too many, try 2.

256 x 2 = 512, so 2 groups of 256 are needed.

16^3	16^2	16^1	16^0
4096	256	16	1
	256 x 2		
	2		

OR
634 ÷ 256 = **2** remainder 122

634 – 512 = 122, so you still need to find groups of place values to make up 122. The next place value available is 16.

122/16 = 7 remainder 10. Therefore 7 groups of 16 are needed. Since the next place value is 1, the value of 10 (the symbol A) should be entered in the table.

16^3	16^2	16^1	16^0
4096	256	16	1
	256 x 2	16x7	10 x 1
	2	**7**	**A**

Therefore the hexadecimal value of 634 is $27A_{16}$

To double-check if your conversion is correct, you can now convert the hexadecimal value $27A_{16}$ to decimal:

2 x 256 + **7** x 16 + **10** x 1 = 512 + 112 + 10 = 634

Method 2

In this method continuously divide the number (634) by the base number (16). Write down the remainder of each division.

634 ÷ 16 = 39 remainder 10
39 ÷ 16 = 2 remainder 7
2 ÷ 16 = 0 remainder 2

answer is $27A_{16}$ (10 = A)

Please note: The answer is obtained by writing down the remainder of each division from the last to the first.

4 ACtivity • Apply your knowledge

1. Write the following decimal numbers in hexadecimal notation.

   a) 20 b) 37 c) 120 d) 290 e) 1456 f) 1200

2. Write your cell phone number in HEX, e.g. 0712345678 = $2A75884E_{16}$

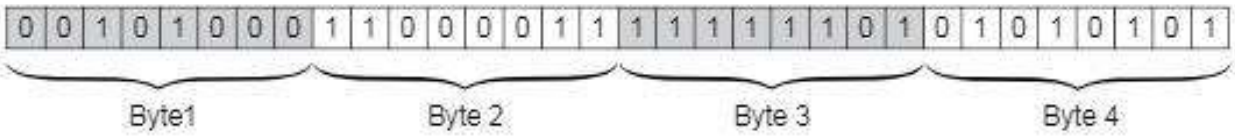
Data storage

While every programming language uses a different scheme of storing data, there are a number of concepts that are common to all of them.

A characteristic of virtually all programming languages is that they are 'typed'. This means that each variable can only store particular types of data e.g. text or integer. Variables of these types are each allocated a fixed amount of memory space determining the maximum and minimum values that each of these variables can store.

Bits and bytes

A byte consists of 8 binary digits or bits, and is a number in binary format e.g. 10110011_2. Each data type uses a fixed number of bytes and data cannot be stored in units of less than 8 bits (1 byte). Here is an example of data stored in 4 bytes of 8 bits each, i.e. 32 bits.



Primitive data types and ranges

Each programming language has variables that we declare or use to store a particular type of data. Each data type usually offers a fixed number of bytes as determined by the language. The three common types used across most languages are:

- Integer or whole number
- Real or decimal or floating point
- String or text

Because each data type is allocated a fixed amount of space (bytes) to store its associated data, there is obviously a limit on the 'size' or range of data that can be stored – the more bytes allocated, the larger the range, and the fewer bytes allocated, the smaller the range.

Here is a table of some integer types, the number of bytes they are allocated and the range of values they can store in the Delphi and Java programming languages:

Delphi			Java		
Data type	*Bytes allocated*	*Range*	*Data type*	*Bytes allocated*	*Range*
byte	1	0 .. 255	byte	1	-128 .. 127
smallint	2	-32768 .. 32767	short	2	-32768 .. 32767
integer	4	-2147483648 .. 2147483647	int	4	-2147483648 .. 2147483647
int64	8	-9223372036854775808 .. 9223372036854775807	long	8	-9223372036854775808 .. 9223372036854775807

Historically, programmers needed to work with the data type which used the least number of bytes, due to memory and storage considerations. Although this is not quite so important today, it is still a good idea not to use unnecessary storage or memory by using data types that allow for much bigger ranges of values than you will ever need. For example, you do not need a 16-byte data type to store the grade of a learner (1–12).

Reliability of data

Variables are allocated a certain number of bytes of space in memory to store values. Therefore problems are experienced when we try to allocate values to the variables that cannot 'fit' into the space provided. This generally leads to unreliable data. These can be divided into two broad categories namely *truncation* and *overflow*, and a *loss of accuracy*.

Truncation and overflow

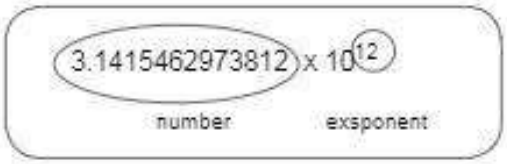
Imagine we had a text or string variable that could store a maximum of 5 characters and we assigned the value 'Addendum' to the variable.

A	d	d	e	n

In this case some of the text will be 'lost' or '***truncated***'. Now imagine the consequences if this had been a number! We refer to the situation where an integer number is 'misrepresented' due to an insufficient number of bytes being available, as ***overflow***.

Loss of accuracy

While programs/programming languages use different ways and number of bytes to store numbers, decimal or real numbers are normally stored in two parts, namely a 'number' part and an 'exponent' part.



Obviously, a loss of accuracy in the exponent part would be critical, whereas a loss of accuracy in the number part would lead to a loss of accuracy in the number itself or in the number of decimal digits.

These 'errors' occur wherever data has to be stored in a set number of bytes. For example, Microsoft Excel and LibreOffice Calc returns 9223372036854780000 if you use the formula =POWER(2,63). However, the correct value is 9223372036854775808.

Storing non-numerical data

Coding schemes

While we have looked at numerical representation, it is also possible to store text or characters by giving each character a unique numerical code. The ASCII (American Standard Code for Information Interchange) system was the original standard which assigned numeric values to letters, digits, punctuation marks, and other characters. For example, the character 'ê', which is often used in Afrikaans, has an ASCII code value of 136. (You can insert the character by holding down the Alt key and typing 136 on the numeric keypad of your computer.)

A good example of where the number of bytes allocated to store data affects the range of 'values', is the ASCII system. ASCII was originally limited to 256 different characters, as only one byte (8 bits) was originally assigned to represent each code. It soon became clear that there was a need for a larger, more universal set of characters, and so UTF-8 and Unicode were developed. These standards were designed to allow for a larger range of 'codes' and hence more characters to be represented.

Pictures and sound clips

Pictures and sound can be represented by a (large enough) series of numbers. If you think of a picture as a collection of thousands of dots, each of which can be modelled by representing its position and colour etc., then we can digitise any picture or video. Music too can be modelled by representing the data as numerical values describing volume, pitch and frequency etc. That's why we talk about the 'digital era' where virtually everything can be represented, edited and transmitted as a collection of numbers.

5 ACtivity • Know the basics

1. Match each of the descriptions in column A with a term from column B.

Column A		Column B	
a)	A number system that makes use of the digits 0-9 and the letters A-F.	A.	ASCII
b)	A collection or grouping of 8 bits.	B.	Binary
c)	A number system that makes use of 8 unique digits.	C.	Unicode
d)	A term used to describe programming languages where variables can only store particular kinds of data.	D.	2721
e)	The number system used to physically store data in computer systems.	E.	Octal
f)	A coding scheme which was originally limited to representing 256 different characters.	F.	Downsizing
g)	An error condition that occurs when there are not enough bits to represent an integer variable.	G.	21
h)	The decimal equivalent of $AA1_{16}$.	H.	Hexadecimal
i)	A coding scheme used that defines a very large set of letters, digits and symbols.	I.	Binary age
j)	A term used to describe a world where virtually everything can be represented, edited and transmitted as a collection of 'numbers'.	J.	Byte
		K.	Typed
		L.	2560
		M.	Digital era
		N.	Overflow

2. Explain what we mean when we say a programming language is 'typed'. Give three common data types used in most programming languages to support your answer.

3. Explain what a bit is in computer terms and what the relationship between a bit and a byte is.

4. Briefly explain why the hexadecimal system makes use of letters such as B and F by referring to a suitable example.

5. The two common errors that can occur in terms of accurately storing data are truncation and overflow, and loss of accuracy. Explain in simple terms what each of these concepts mean by referring to suitable examples.

6. What are ASCII codes?

7. Why was the ASCII system extended to Unicode?

6 Activity • Apply your knowledge

1. In general, how does the number of bytes allocated to a particular data type affect the range of values that can be stored in a variable of that particular data type?
2. Ultimately, computers represent and store data in terms of numbers. Briefly explain how this is possible in terms of characters such as '$' and images.

7 Activity • Think and research

1. Write down the list of valid 'digits' that could be used in a base 6 number system.
2. What is the equivalent value of the number 105_6 in a decimal system?
3. Briefly motivate why the binary system is so important in terms of computers.
4. A new programming language called SQWERTY has been developed. It uses three bytes (of 8 bits each) to represent an integer value in binary. The left-most bit is used to indicate if the number is positive or negative. Work out what the largest integer value is that you can store in an integer variable in SQWERTY.

Glossary

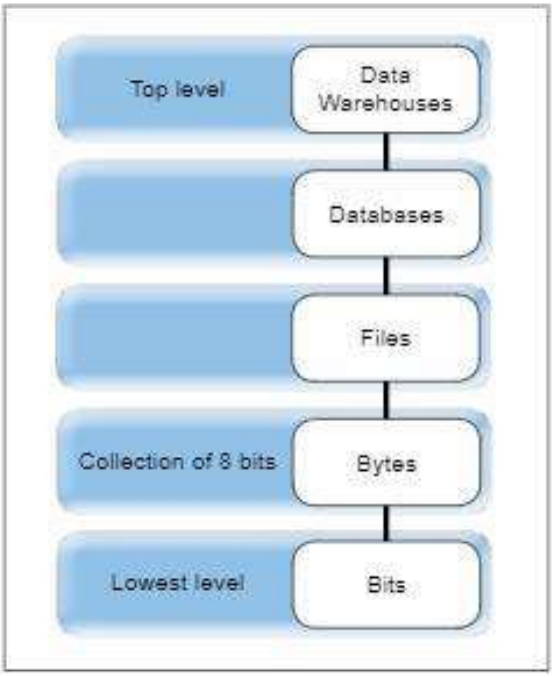
ASCII	ASCII (American Standard Code for Information Interchange) is a coding system which is limited to 256 different characters.
Binary	The base 2 number system that consists of just two digits, namely a 0 and a 1.
Bit	A binary digit i.e. a 0 or a 1.
Byte	A collection of 8 bits – the smallest unit of storage in a computer system.
Coding scheme	Schemes used to represent or store text or characters by giving each character a unique numerical code.
Decimal	The base 10 number system that consists of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.
Hexadecimal	The base 16 number system that consists of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F.
Octal	The base 8 number system that consists of the digits 0, 1, 2, 3, 4, 5, 6 and 7.
Overflow	An error condition that occurs when there are not enough bits available to represent an integer value accurately.
Primitive data type	The core types of a programming language. Variables are declared and used to store a particular type of data.
Truncation	The misrepresentation of a string variable because there are not enough bytes available to store all the characters in the string.

MODULE 4.2

File management

Introduction

The way data is represented in the computer can be seen as a hierarchy. The lowest level is the level at which the machine works, i.e. bits and bytes. In Module 4.1 we looked at this level.



The highest level is that of a data warehouse. A data warehouse is a vast collection of data gathered from a variety of databases and other data sources. It is specifically designed to determine trends and patterns within the data and it is designed for reporting and querying.

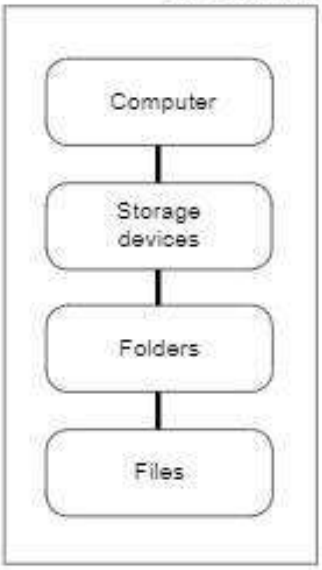
In this module, we will study the level of data representation that most of us will work with, namely *files*.

A computer stores data in electronic format in its memory while the computer is switched on. Once the computer is switched off, all data in memory is gone. Therefore files need to be saved permanently so that we can access them at later stages, on some form of storage device such as a hard drive, CD and DVD disc and flash drive.

In addition, files allow us to exchange information and data with one another whether it be posting a photo on Facebook or sending a file via e-mail.

The file manager

To see which files are stored where, we make use of a file manager program that forms part of the operating system. For Windows it is Windows Explorer. For other operating systems such as Linux, Nautilus can be used. These file managers can provide information on files as well as the ability to manipulate the files.



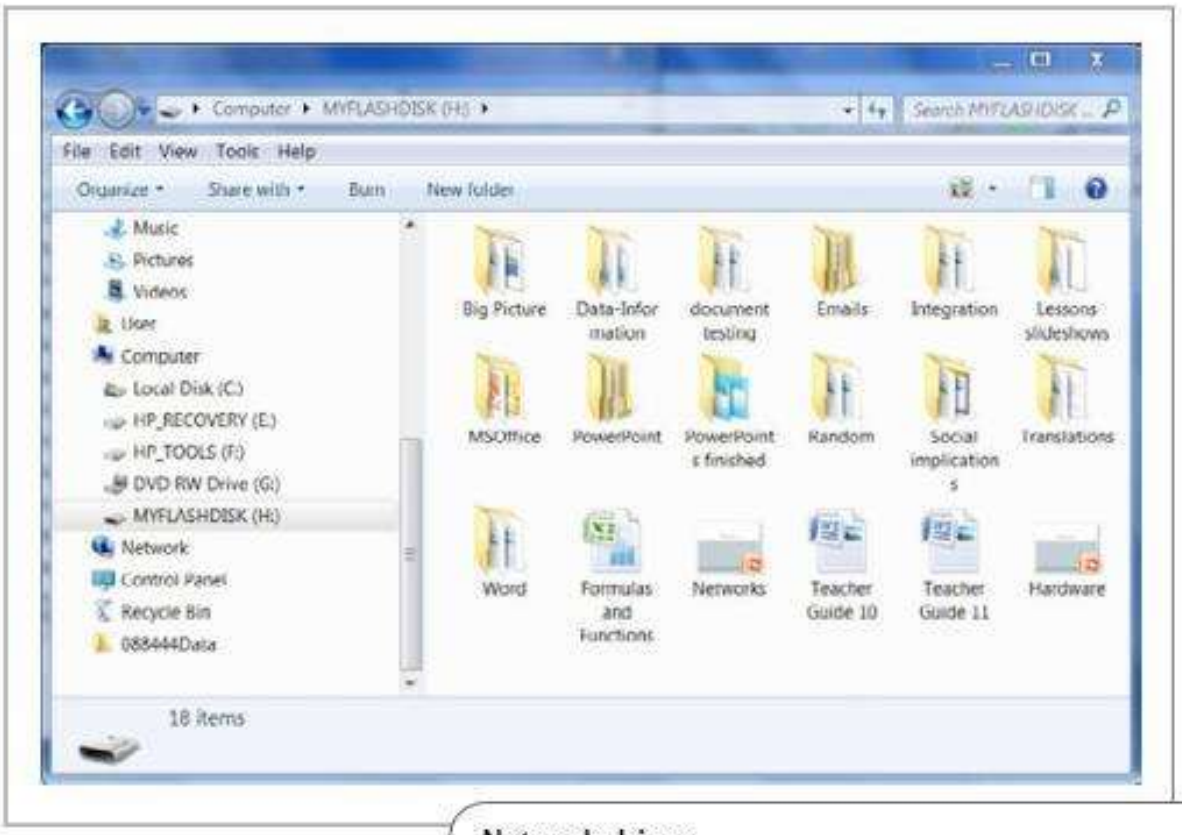
In this module, we will use Windows Explorer as an example of a file management program.

Storage hierarchy

A diagram of the hierarchy for storing files is given.

Windows Explorer provides information on all the storage devices attached to a computer. These devices are called **drives**. The storage devices could be for example the hard drive, a flash disk and a DVD drive. (A hard disk can be divided into more than one partition, and each partition is also called a drive. Each drive is named using an alphabet letter e.g. C: or F:.)

On the drives **files** can be grouped in separate containers called **folders**. A folder may also contain other folders, which we then term subfolders.

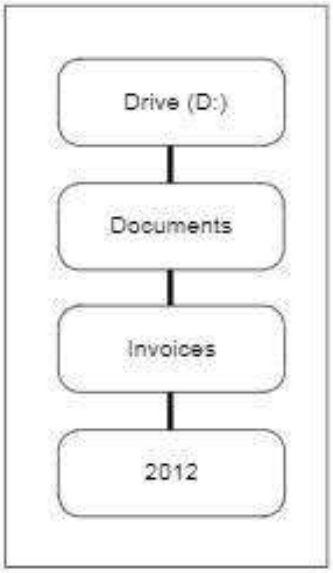


Network drives

You will see other 'network drives' if your computer is connected to a network. These are usually designated with letters lower in the alphabet such as S, T and U. These are in fact not physical drives but folders situated somewhere on another computer of the network that act as, and give the appearance of being physical drives. In this way for example, all users can see the same 'drive' each for their own separate storage and work area. In other words, all users might have a U-drive where they can save their work, but each U-drive is different as it is in fact a different folder for each user's work.

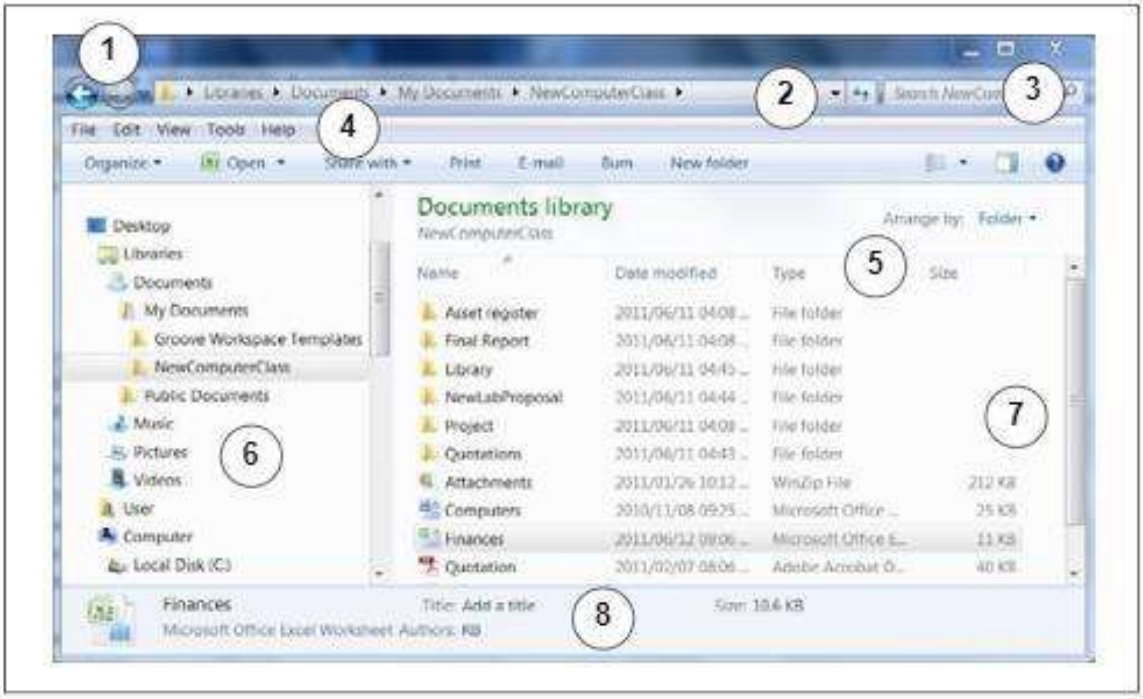
Paths

- A path is the 'succession' of a drive and folders that lead to the location of a specific file. The backslash character is used to separate the 'list' of folders in the path. For example, the path to a specific file might be D:\Documents\Invoices\2012 (meaning that the file is stored in the 2012 folder, in the Invoices folder, in the Documents folder).
- Sometimes the name of the computer on which the file is stored is also given e.g. \\Server1\Backup\2012\ meaning the folders are located on the computer named Server1.
- The path of a file is always displayed in a bar at the top of a file manager.



Working with Windows Explorer

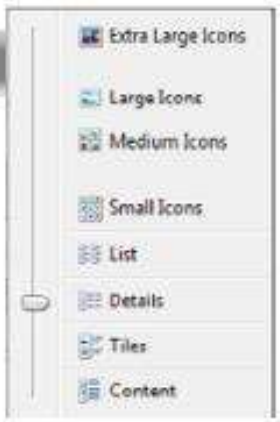
Windows Explorer is accessed via the Start menu or by pressing the Windows (⊞) and <E> keys simultaneously. In Windows Explorer, you will notice that the screen is divided vertically into sections. We refer to this as the left or navigation pane and the right pane respectively:



1	**Back and Forward buttons**	These are used to navigate between folders you have already opened.
2	**Address bar**	Shows the 'path' to the current folder and is used to navigate to a different folder by either typing or clicking another folder on the Address bar.
3	**Search box**	You can enter a word or a phrase in the Search box to look for a file.
4	**Toolbar and menu bar**	These allow you to perform common tasks, such as changing the appearance or 'view', printing files and copying files to a CD.
5	**Details or column headings**	You can choose which details about the files you wish to display. The contents of the folder can quickly be sorted in ascending order by clicking on the appropriate column heading.
6	**Left (Navigation) pane**	This is where you can scroll up or down to other folders.
7	**Right pane**	This is where the contents of the current folder are shown.
8	**Details pane**	This is where some of the properties of the selected file or folder are displayed.

1 Activity

- Open Windows Explorer.
- Experiment with the different views that are available under the Views menu option.
- You will notice that some of the folders have a ▷ sign or symbol in front of them which indicates that there are sub-folders.
- Click on one of the ▷ signs. Look what happens in the left pane. (Nothing changes in the right pane.)



- Now click on a folder name in the left pane and note how the contents of the folder are displayed in the right pane.
- Browse through the contents of some of the items in the left pane by clicking on them.
- Once you open a few of the items in the left pane, you will notice that some of them suddenly have a ◢ sign front of them. Click on a ◢ sign and notice how the folder structure 'collapses'.

The unit in which we measure the size of a computer file is the Kilobyte (KB). You will see that the size of files in Windows Explorer is shown in terms of KB. One byte is generally accepted as the space required to store a single character.

- 1024 (approximately 1000) bytes is called one Kilobyte (1 KB)
- 1024 (approximately 1000) KB = 1 Megabyte (1 MB)
- 1024 (approximately 1000) MB = 1 Gigabyte (1 GB)
- 1024 (approximately 1000) GB = 1 Terabyte (1 TB)

Files – the basic unit of storage

A file is a collection of data (bits) that is stored on a storage device under a single name. A file can for example be a letter you typed or a photo you took with a digital camera.

Each operating system has its own way of dealing with files and how they can be saved. To start, we need to look at how files are named in Windows and familiarise ourselves with common file types we are likely to encounter.

File name conventions

Every operating system has its own set of rules in terms of how file names can be defined. In addition, we can set our own conventions of naming files so that they are easier to organise.

Operating system defined rules for file names

File names in the Windows operating system consist of two parts, namely the file name, followed by a period (full stop) and a 3–5 letter file extension. Examples are Data.accdb (an *Access* file) and a program file such as Delphi.exe. The file extension identifies the file type or format and is automatically added by the program when you save the file. For example, *Word* will have all its files with an extension of .docx.

Tip:
The safest bet is to always use just letters and digits for file names if you want them to be compatible with all operating systems.

Windows limits file names to a maximum of 260 characters. This, however, includes the 'path' where the file is kept e.g. C:\My Documents\Games\Cracks.html. You can use any characters in the file name except the following ones: \ / ? : * " > < |

The Linux operating system generally places fewer restrictions on its file names than Windows does but the file names are case sensitive in some versions.

Significance of file extensions

Operating systems use file extensions to identify which program to use to open the particular type of file. For example, a plain text file (.txt) might be set to be opened by the Notepad or the WordPad text editor utility.

It is generally not a good idea to change the file extension yourself as Windows may no longer

recognise the file type and therefore not know what program to use to open the file. Windows will, by default, hide (not show) the file extensions of file types so that a user cannot change or delete the extension by mistake.

If an operating system encounters a file type that it does not recognise, it will give a list of suggested programs to try to open the file or allow you to search the Internet to find the suggested program to open it. You can also change or set the default program that must be used to open a particular type of file.

User defined conventions

Individuals and organisations may decide on a naming convention for their files, to be consistently named e.g. *Report 01 2012* and *Report 02 2012*. This will make it easier to find the files they want to access.

Common file types and extensions

Although the file extensions are not normally displayed, we often refer to the different types of files by their file types. For example, we might refer to all the 'jpeg files', meaning graphics files that are stored using the jpeg format.

The following is a collection of some common files (file types) as shown by a file manager program (each file is represented by an icon).

The following table provides a list of the more common file types you are likely to encounter.

File type	File extension	Program or type of program associated with the file
Graphics files	*jpeg, bmp, gif, wmf, tiff, png*	Any graphics program (including *Paint*). These files can be inserted into a file by most programs such as *Word*, etc.
Office	*docx, xlsx, pptx, accdb*	Microsoft Office suites files for word processing *(Word)*, spreadsheet *(Excel)*, presentations *(PowerPoint)* and database *(Access)*. Open Office opens files with the extensions *odt, ods, odp* and *odb* respectively.
Audio files	*mp3, mp4, wav, cda, wma*	Any media player program such as Windows Media Player.
Video files	*mpeg, asf, wmv, avi, mov, flv*	Any media player program such as Windows Media Player or video editing software.
Portable Document Format	*pdf*	You just need a utility program called Adobe Reader to read it. This is a file format often used to distribute document files, as it does not require the program in which the document was created to be installed to read it. The document cannot be edited unless you have the Adobe writer program.

File type	File extension	Program or type of program associated with the file
Compressed	*zip* of *rar*	Files or folders can be compressed to save space. The files in the compressed or 'zipped' folder have to be decompressed or 'unzipped' in order for one to open them and use them, using a program such as Winzip or Rar.
Hypertext Markup Language	*html*	A web page which can be opened in a web browser. HTML is used to define the structure and layout of a web page.
Font files	*otf, ttf*	A font is a particular design or 'appearance' of how characters appear when using that particular type of font. The font refers to the specific typeface e.g. Arial or Calibri and the size and spacing between characters. The two most common font file extensions are OTF (OpenType fonts) and TTF (for TrueType fonts). An operating system like Windows comes supplied with hundreds of font files. Additional font types can be downloaded (some for free) from the Internet.
Plain text files	*txt*	A plain text file contains lines of plain text. The file extension txt is associated with text files but a number of other text file extensions also exist. For example, the source code for computer programs is often stored in text files with extensions such as *pas* (Delphi) and *java* (Java). Text files can be opened by any word processing application or simple text editing utilities such as WordPad and Notepad. They are often used to quickly and easily store information because of their relative ease of use.
Rich Text Format	*rtf*	RTF files support text formatting, such as bold and italics, fonts and font sizes, page setup and tab settings and even images. This is a useful format to use as it is universally used, supported by most word processors.

This is not by any means meant to be a complete list of every file type you will come across, but rather examples of the more common ones you are likely to encounter. You will be creating your own programs in IT, so let's spend a bit of time talking about program files.

Program files

Programs range from custom specific programs such as one designed to control the traffic lights at an intersection, to more generic programs such as the Microsoft Office applications, which allow the user to create a variety of tasks. Programs very seldom consist of one single file, which is why there are special procedures needed to install programs.

A program file that can be run or executed generally has a file extension of *.exe* meaning that it is an executable file. How are these files created?

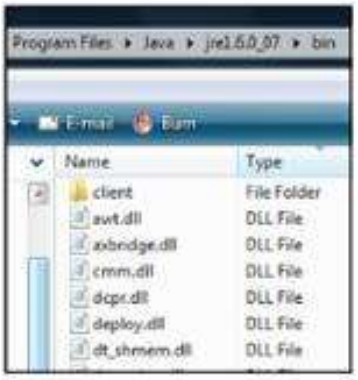
Source code and object code (executable file)

Various programming languages such as Delphi, Java, C# and Python exist. Each of these allows one to construct and specify instructions in order to create a set of programming instructions or source code. These instructions can be sent to a program called a compiler, which checks to see that the code adheres to all the rules of the programming language and creates object code (an executable file) which can be 'run' or executed.

Example of Delphi programming code	*Example of Java programming code*

`iNum1 := StrToInt (edtNum1.Text);`
`iNum2 := StrToInt (edtNum2.Text);`
 	
`iSum := iNum1 + iNum2;`
`lblOutPut.Caption := IntToStr(iSum);`
	`number1 = input.nextInt();`
	`number2 = input.nextInt();`
	
	`sum = number1 + number2;`
	`System.out.printf( "Sum= %d\n", sum );`

Shared and dynamically linked libraries



As mentioned previously, programs seldom consist of one single file. In addition, a lot of code or instructions are common to many, if not all programs.

The operating system will for example provide 'libraries' or collections of code that allow you to perform common tasks such as displaying output on the screen.

Programmers place these common or shared functions in files called DLL or Dynamic Linked Libraries. These routines can then be called by programs needing to run any of the routines from the library.

The other advantage of having shared 'libraries' of programming routines is that it is a lot quicker and easier to update a program if only a DLL file needs to be updated instead of the whole program.

Converting between file types

While each program has its own unique format in terms of representing data in each file created by the program, it is often advantageous to be able to save the file in a format that can be read by other types of programs. This is often done to

- share data by using a common format that both programs can read and interpret.
- allow users who are using a different operating system to access the data.
- allow users who are using older versions of a program to access to the data.

2 Activity

Open a blank document in your word processing application (e.g. *Word*) and perform a **Save As** command. Just under the **File name** box, you will see a drop list control with the name **Save as type**. Activate this control by clicking on the drop-down arrow and examine the various options presented.

Some of the options allow you to save the document in a 'older' (previous) version, others options allow you to save it as a web page or as a plain text file (without any formatting).

3 Activity • Apply your knowledge

1. Explain what is meant by the term the 'data hierarchy' with the aid of a sketch.
2. An icon of a compressed or 'zipped' file is shown
   a) Explain what the term 'zipping a file' means.
   b) Give the main reason why one would perform this type of operation.
   c) What does one have to do before you can work with a file or folder that has been 'zipped' or compressed?
3. Give two limitations some operating systems place on the naming of files.
4. What program or type of program would an operating system most likely try and use to open the following files? Briefly explain your answers.
   a) Agenda.pdf
   b) Agenda.txt
   c) Agenda.zip
   d) Agenda.odt
   e) Agenda.docx
   f) Agenda.exe
   g) Agenda.jpeg
   h) Agenda.mp3

5. While trying to rename a file, a user was presented with the following error message:

   a) Explain what a file extension is by referring to suitable example.

   b) Explain what is meant by this error message when it says 'the file may become unusable' by referring to the significance that a file extension plays in an operating system.

6. Give two reasons why it may sometimes be necessary to store data in a different file format other than the default one used by a particular program.

7. Your father says that he noticed that he can open all of his spreadsheets by first opening Excel itself, then executing the command to open a file. When he double-clicks on the spreadsheet's name in a file manager, it automatically opens in the Open Office program called Calc. Explain to him what the likely cause of the problem is and how to fix this problem.

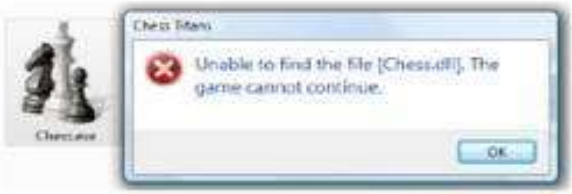
8. Describe in broad terms how a program is 'created' by referring to the terms source code, object code and compiler in your answer.

9. A user was trying to run the Chess program that comes with Windows when she encountered the following error message:

   a) Briefly describe why this error message has occurred by describing what a DLL (Dynamically Linked Library) file is.

   b) Give two advantages that DLL files provide.

   c) Can you suggest a plan of action to solve this problem?



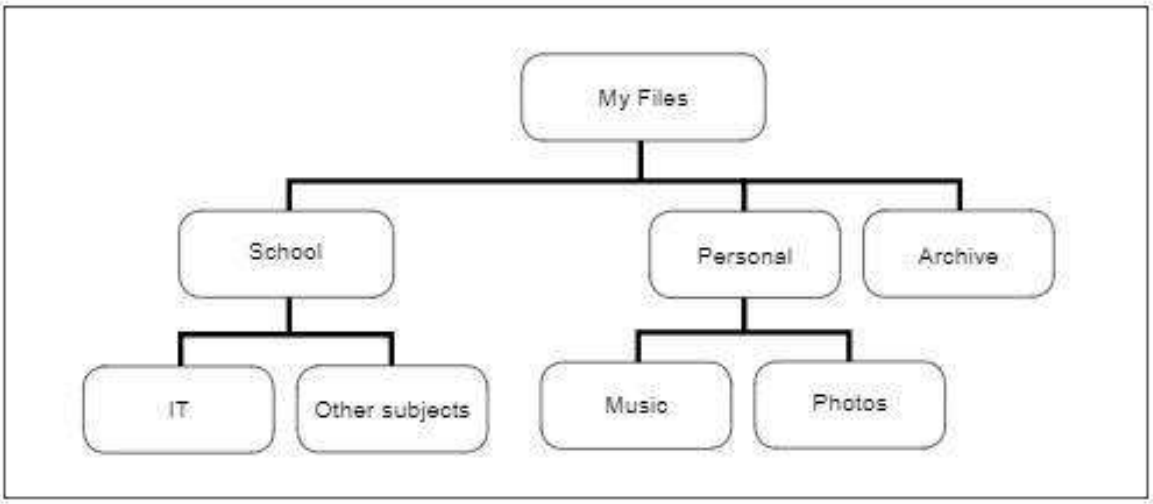
The need to organise

Because we can store so many files (particularly on a hard drive), we need to be able to organise these electronic files just as we would have to organise a manual filing system for it to be effective.

Plan a folder structure

Always plan (and maintain) a logical folder structure so that it is easy to locate and manage your files. Here are some general tips:

- Think of the main 'categories' you would need to save your files under e.g. School and Personal as shown in the diagram. Now keep on refining each level.



- Make a logical decision when to stop refining – it's pointless having a 'tree' or folder structure so complex that it is difficult to navigate and you end up having only one or two files in each folder!
- Keep on refining and maintaining the structure once it has been created. If you find you start having too many files in one folder, it might be an indication that you should sub-divide the folder by adding sub-folders. As an example you might create separate subfolders for your programs and other work for IT.
- Use meaningful names for your folders and files so that they are easily identifiable.
- Possibly move files that you seldom use into a compressed file or store them in another folder such as the Archive folder shown in the diagram. Some users like to keep the same folder structure but stored under different years, 2012, 2013, 2014, etc.

Managing files and folders

File management software can be used to implement and maintain the folder structure as well as perform operations on files and folders such as

- copying and cutting (moving)
- renaming
- searching
- deleting
- compressing.

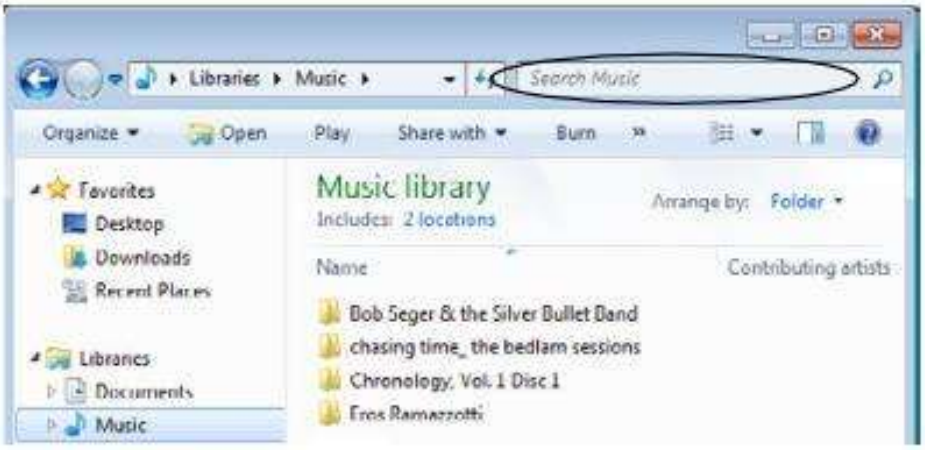
Finding files in Windows

With potentially thousands of files on the hard drive, it can be quite a nightmare to locate a file that you know you have saved but can't remember where! Windows provides very powerful search facilities to locate these files. You can search using a combination of criteria. These include

- the file name (or part of it)
- some text you know it may contain
- the drive or directory to look in
- when the file was modified
- the size of the file
- the approximate size of the file.

Obviously the more information you can supply, the easier it will be to narrow down the search and locate the file.

Different search facilities are available in different operating systems. However, for all of them you can enter data (such as the file name) as criteria for the search.



Using wildcards

Windows allows us to use an asterisk (*) or a question mark (?) to represent one or more characters when searching for files or folders. These are known as wildcard characters and can be used in place of one or more characters when we do not know what the real character is.

We use the asterisk as a wildcard character for zero or more characters. For example, if we are looking for a file we know starts with 'tel' but cannot remember the rest of the file name, we can type the following: tel*

This would locate all files of any file type that begin with 'tel' including telephone.txt, telling.docx and tell me more.xlsx.

To narrow the search to a specific type of file, we can specify the file extension: tel*.docx

This would locate all files that begin with 'tel' but have the file name extension docx, such as telling.docx.

We can also use the question mark as a substitute for a single character in a name. For example, if we type tel?.docx, we could locate the file tel1.docx or tell.docx but not telephone.docx.

4 Activity

In this activity you are going to practice specifying different criteria to search for files using Windows Explorer. The more information you supply, the more likely that your search will be successful. The folder Search Data is supplied.

Study the file SearchFacility to find the specific instructions for your operating system.

Try the following searches using combinations of search criteria:

- Find all *Excel* files.
- Find all *GIF* files containing the word Black in the file name.
- Find all files containing the word supply.
- Find all *Excel* files that are least 18 KB in size.
- Find all files modified in the year 2001.
- Find all *Word* files modified in the 1990's that are more than 50 KB in size that have the word 'pet' as part of the file name.

5 Activity

As schools are using a wide variety of types and versions of operating systems, we cannot give a detailed set of instructions on the way your file manager works. Your teacher will be able to assist you, but you need to ensure that you are able to perform all these functions – tick these off as you master them:

1.	Create a new folder	
2.	Copy, move, delete and rename files and folders	
3.	Navigate to a specific folder to retrieve or store a file	
4.	Search for a file based on its name, properties or contents	
5.	Open files and folders	

6 ACTivity • Apply your knowledge

1. Your disk is getting very full. A friend suggested that you save space by compressing or zipping all the files on the disk. Comment critically on this advice.
2. What program or type of program would an operating system most likely try and use to open the following files? Briefly explain your answers.
   a) Afile.mov
   b) Afile.rtf
   c) Afile.flv
   d) Afile.zip
   e) Afile.html
3. Explain why some computers seem to have 'extra drives' with labels such as U: or S:
4. What type of operating system utility do we use to manage our files and folders?
5. What is a hard drive partition?
6. Give five tips or issues one should consider when planning a folder structure.
7. What is the difference between a path and a file name? Illustrate your answer with a suitable example.
8. What is the name of the file manager tool that is supplied with the Windows operating system?
9. What type of details (besides the file name) are usually given about a file when using a file manager?
10. What is the difference between the * and the ? wildcard characters in Windows?
11. What file would be found by the following search string: ?.txt

Glossary

Audio files	Files that contain sound or music. Common audio file formats include MP3, MP4, WAV, CDA, WMA.
Compiler	A program that checks to see that the code adheres to all the rules of the programming language and creates an executable file which can be 'run' or executed.
Compressed files	A file format used for files that contain files or folders in a compressed format to save space. Typical file extensions include rar and zip.
Data hierarchy	Refers to the two levels of data storage – the lower bits and bytes level and the higher levels of files, databases and data warehouses.
Data warehouse	A data warehouse is a vast collection of data gathered from a variety of databases and other data sources. It is specifically designed to determine trends and patterns within the data and it is designed for reporting and querying.
DLL files	DLL or Dynamic Link Libraries are files that contain code or routines that can be called and used by programs.
File	A file is a collection of data or information that is stored on a computer under a single name. They are represented as icons on the computer.

File extension	This set of characters (usually 3-5) is added to the end of a file name and it identifies the type of file e.g. docx is the file extension of a Microsoft Word file.
File format	How data is structured or organised within a file. Each program has its own file structure for its files. Some programs can interpret or read files that have a different file structure to the one they use.
File manager	A utility program like Windows Explorer that comes as part of the operating system. It allows the user to manage their files and folders.
File name	Name given to a file followed by a full stop and the file extension e.g. MyFile.docx.
File-naming conventions	Rules and restrictions that the operating system places on how a file can be named. Organisations or individuals might also choose to name their files according to a fixed format that they choose.
Folder	A folder is a storage place or container in which you can store files or more folders on disk.
Font files	Files containing fonts that can be used by programs on the computer where the fonts are installed. They typically have a file extension of TTF or OTF.
Graphics files	Files that contain images or graphics. Common graphics formats include JPEG, BMP, GIF, WMF, TIF, PNG.
HTML files	Hypertext Markup Language files are is used to define the structure and layout of a web page.
Hypertext Markup Language files	Hypertext Markup Language files are is used to define the structure and layout of a web page.
Path	The 'sequence' of folders that lead to the location of a specific file. The backslash character is used to separate the 'list' of folders in the path e.g. D:\Documents\Invoices\2012.
PDF files	Portable Document Format files are used to distribute document files, as they do not require the program in which the document was created to be installed to read it. You just need is a free utility program called Adobe Reader to read it.
Plain text files	A text file that contains lines of plain text. The file extension .txt is associated with text files Text files can be opened by any word processing application or simple text editing utilities such as WordPad and Notepad.
Program files	These are actual programs, not data files, that run (execute) when opened. They normally have a .COM or .EXE file extension.
Programming Languages	Programs such as Delphi, Java and C# that allow one to construct and specify a set of programming instructions or source code.
Rich Text files	Rich Text format files support 'rich text' which includes text formatting, such as bold and italics, fonts and font sizes, page setup and tab settings and even images.
Wildcards	Characters that represent one or more characters. For example Windows allows us to use an asterisk (*) to represent any number of characters or a question mark (?) to represent one single character when searching for files or folders.

Section 5

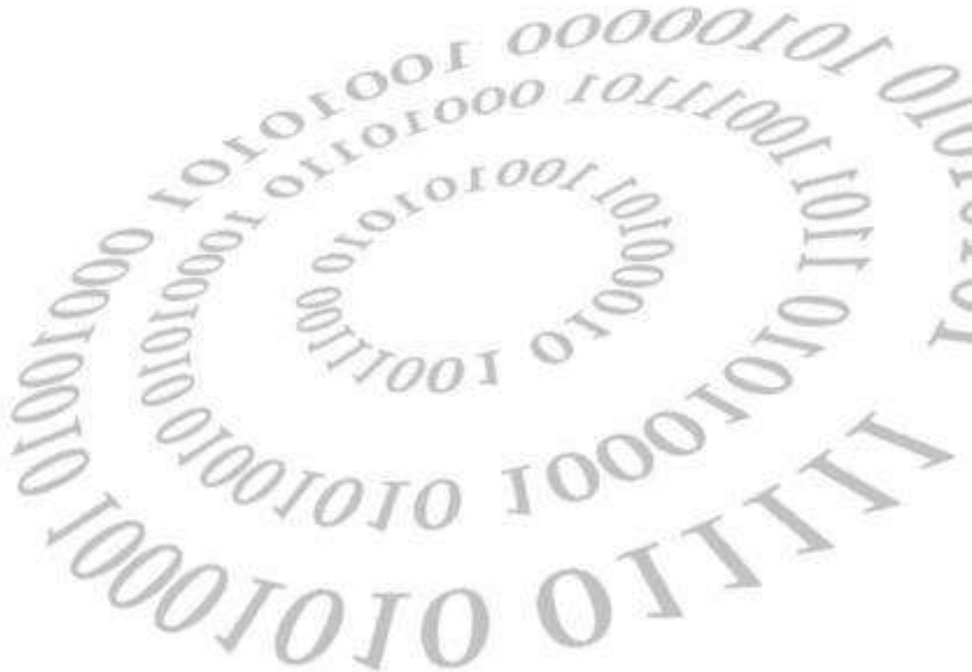
Social Implications

Modules

Module 5.1 Social, ethical and legal issues surrounding ICTs

Module 5.2 Green, health and global e-communication issues

Module 5.3 Safe Internet and e-mail use



MODULE 5.1

Social, ethical and legal issues surrounding ICTs

Introduction

We've been spending a lot of time looking at technical details about the insides of computers and software and learning to program. You also need to learn to place ICT into context by thinking about the impact and effect that ICT has on people, our society and culture.

Did you know that every time you click on 'I accept' on a dialog box you are signing a legal contract? Did you ever stop to think that leaving your computer on all night to download a file has an environmental impact simply because of the power it uses? Did you know that some people have to have surgery to fix problems created by using computers? Is it true that the world is split into two camps by technology?

We should never think about technology only in terms of the job it does – we also need to think of it in terms of the impact it has on the world and on humanity.

As you go through your three years of ICT at school we will highlight some of the social issues relating to the work that we have covered and to the use of computers in general. This is only the tip of the iceberg in this area of study and we encourage you to keep your eyes and ears open and think about the moral, ethical and social impact of ICT instead of just being a blind user!

Let's start off by looking at a couple of the legal issues related to ICT and its use.

EULAs, Copyright and piracy

EULAs

The EULA is not some strange war cry – it stands for End User License Agreement. What it means is that you never own any software that you buy, no matter how much money you spent on it. All you get for your money is a *license* to *use* the software, usually on only one computer.

The EULA is that long document that appears whenever you install software. Generally, you have to click on an 'I Accept' button before you can install the software. That document, which most of you have probably never read, is a legally binding contract. You 'sign' it by clicking on the 'I Accept' button. EULAs tend to be completely biased in the software company's favour. Their software can be buggy, crash, destroy your computer or even cause the end of the world (ok, we're exaggerating here) and you cannot hold them accountable or sue them. All that they are responsible for is replacing the media if it is faulty!

The software company, on the other hand, can sue you if you install on more computers than your license allows. Some EULAs even allow the software company to come and search your computer without a search warrant!

Copyright

Copyright is designed to protect intellectual property. Intellectual property is anything created by the mind. Copyright is used to protect things such as photos, music, art, movies, books – and software. What all these types of work have in common is that it is very easy to make and distribute perfect copies of the work at virtually no cost – especially when the work is in a digital format.

This makes this type of work very easy to steal. You can't make a copy of a TV set without spending a lot of time and money, buying parts and using expensive equipment. However, you can a make a copy of a movie, book or computer program that took many people and a lot of time and money to create. Producing your copy can cost you virtually nothing, but this does not mean that you are not stealing. Copying intellectual property without paying for it (or getting permission) is just as much a theft as taking a physical object – like a TV set.

Copyright law protects intellectual property by creating legal limits on who can copy, sell and use this type of work. In many cases, simply taking a photo or writing a document, book, poem, etc. means that you automatically have copyright over the work, as long as you can prove that you have created it. Copyright is normally indicated by the word Copyright or the © symbol followed by the name of the copyright owner and the year that the copyright was first applied.

Did you know?

Copyright law is different in different countries. It has a limited lifespan (up to 75 years after the death of the creator) and once copyright has expired, the work falls into the 'public domain'. Public domain works can be copied freely. Project Gutenberg on the Internet is a project that is aimed at creating free e-book versions of books that are in the public domain.

Copyright is not the only law that defends intellectual property – other laws include the whole patent and trade mark system.

Copyleft allows people to use and distribute work whilst still allowing the creator some control over where and how it can be used.

Creative Commons are Copyleft-style copyright licenses that allow you to use parts of, copy and distribute work for non-profit purposes.

GNU / GPL is a license specifically designed for software. It allows people to distribute the software for free or even charge for it so long as the source code is made available. GNU is Copyleft and is the license most freeware is distributed under.

Piracy

Piracy is the act of stealing intellectual property, i.e. using or making a copy of the kind of work protected by copyright.

Piracy includes:

- Buying one license for software and installing it on multiple computers.
- Installing and using software that is not free without paying for it.
- Sharing software with friends and family when you are not permitted to do so.
- Making copies of, sharing or downloading copies of media such as books, music and movies that you have not paid for.
- Selling illegal copies of media or software.
- Large scale copying and selling of illegal counterfeit copies of media or software by pirate businesses.

Piracy is more than just theft. It has wider-ranging ranging social implications than just the act of stealing. Consider the following:

- Other people are forced to pay for your crime when you pirate something. Businesses estimate their losses from piracy and build that into the prices that they charge people – making their products more expensive!
- Piracy also takes away some of the motivation for people to create new products. Would you want to spend a lot of time and money and effort to make something, knowing that people are simply going to steal it and that you won't get the rewards you feel you deserve?
- Pirated goods is big business – but the business is not only illegal, it is not taxed and does not contribute to the economy of a country.

So why do we use Computers anyway?

Computers are fun to use and allow us to be more productive and creative. Here are some economic reasons why we use computers and ICTs:

- Computers are generally more accurate and reliable than people (they don't get sick, tired or bored).
- They are fast – we can do more tasks in the same time by using a computer.
- They can work 24 hours a day, 7 days a week.
- They cost less than people in the long run (salaries have to be paid continually, but the running cost of a computer – after the initial purchase – is much lower than the salaries of the people it would take to do the same work).
- They make communication faster, easier and cheaper than ever before with technologies such as e-mail, web sites, VoIP and video conferencing.
- They reduce the need for other resources – one can cut down on paper, office furniture, office space, etc.
- A well-designed ICT system can improve the over-all effectiveness and running of a business.

The Digital Divide

William Gibson (a science fiction author who created the term Cyber Space) once said: "*The future is already here – it's just not very evenly distributed.*"

You have just seen that computers are really powerful tools that increase productivity and broaden horizons, and that have definite economic advantages. Can you imagine being someone who has never seen a computer, has no access to ICT and has no idea of how to use ICT or what it can do? Imagine not knowing what the Internet was, or how to use it. That scenario, in a nutshell, describes the digital divide. It is a self-fuelling problem, as citizens who are not 'digitally enabled' cannot always benefit from access to services and information, or from opportunities to better themselves and their situations. In this way the divide or gap can grow rapidly as technology evolves at such a rapid rate.

Factors influencing the Digital Divide

The digital divide occurs when individuals or groups (even countries) do not have access to, or the ability to make use of, technology, as compared to digitally-enabled groups. This gap or divide is, amongst other things caused by:

- A lack of financial resources to purchase or access technology.
  The lack of wide-scale public access to affordable broadband Internet technologies is a major stumbling block in our own country.

- Differences in educational levels.
  The better educated individuals are, the more likely it is that they can make use of technology.
- The age gap.
  Many 'older' people have not had formal training in technology, or they've never needed to use technology in their workplace before.
- Competition for financial resources.
  In developing and third world countries, large sections of government spending is needed for more 'basic' needs, such as health and housing
- Disabilities.
  Disabled individuals often experience problems in terms of making effecting use of ICT due to their sight, hearing or limitations in terms of their cognitive levels or physical mobility.

1 Activity

1. A customer was astounded when the salesperson who sold him a new version of Microsoft Office, remarked to him that he did not own the software, even though he had just 'bought' it!

   Explain to the customer why this is correct by referring to the concept of a software license in your answer.

2. Your mother was installing some software on her laptop. Her boss told her just to click on the 'I Accept' button when the installation program asks her to accept the EULA, as this is not important - it is just a way of checking if you actually want to install the software.
   a) Explain what a EULA is. Do not just give the meaning of the acronym.
   b) Do you agree with the advice that accepting the EULA is not that important, as it is just a way to check if you actually want to install the software? Briefly motivate your answer.

3. Your younger brother made a copy of a friend's game. He maintains that it cannot be theft, as the original copy is still available, and has therefore not been 'stolen'. He has also read that it is perfectly acceptable to do this in some countries!
   a) Explain to your brother why his thinking is not correct by referring to the concepts of intellectual property and copyright.
   b) What term is commonly used to describe software that has been illegally copied and distributed?
   c) Comment on why the illegal copying of software seems to be more 'acceptable' in some countries than in others.

4. You have downloaded some freeware. When installing the software, a pop-up stated that the software had a *GNU / GPL* license. Your dad says that this is a *copyleft*-style copyright licence, issued under the *Creative Commons* banner or umbrella.
   a) What is a GNU / GPL license?
   b) What do we mean when we say a license is copyleft?
   c) What is Creative Commons?

5. Computers and related technologies allow smaller businesses to function more efficiently and effectively, enabling them to compete with larger businesses and competitors across the world.
   a) List three communication technologies that enable companies to communicate with their customers more easily and cheaper than ever before.
   b) Give five reasons why computer technologies (excluding communication technologies) allow a business to operate more efficiently.

6. The Digital Divide is a huge stumbling block to sustainable growth and development in Africa.
   a) Explain what is meant by the Digital Divide.
   b) List five factors that cause or contribute to the Digital Divide.

Glossary

Copyleft	A type of Copyright law which allows people to use and distribute work whilst still allowing the creator some control over where and how it can be used.
Copyright	A law designed to protect intellectual property by creating legal limits on who can copy, sell and use types of work such as photos, documents, books or poems. Copyright is normally indicated by the word Copyright or the © symbol followed by the name of the copyright owner and the year that the copyright was first applied.
Creative Commons	A copyright license that allows you to use parts of, copy and distribute work for non-profit purposes. Creative commons is a Copyleft-style license.
Digital Divide	The gap between people that have access to computers and technology and people that do not.
EULA (End User License Agreement)	A licence agreement (which is a legally binding contract) in which you agree to use software only according to the software company's terms. Most software won't finish installing until you have indicated that you are aware of the terms and conditions of this licence agreement, and that you accept them. These terms and conditions differ from licence to licence, but they mostly involve things such as the number of computers you may install the software on, and whether you may distribute the software or not.
GNU / GPL	Is a license specifically for software that allows people to distribute the software for free or even charge for it so long as the source code is made available. GNU is Copyleft and is the license most freeware is distributed under.
Intellectual property	The idea behind any original creation is the property of its creator, and not only the physical objects which might express that idea. If, for example, you were to write a poem or a book, or take a photo, and someone were to make a copy of it without asking your permission or compensating you, that person would have stolen your intellectual property, even if he didn't steal your physical exemplar of the work.
Piracy	The act of stealing Intellectual property. i.e. using or making a copy of the kind of work protected by copyright law.

MODULE 5.2

Green, health and global e-Communication issues

Computers are not the magical wonder tool that solve all the world's problems and have no faults of their own. ICT raises many issues of its own.

Ergonomics and health issues

We were not meant to live the kind of desk-bound life we do in the information age. The lack of physical activity caused by a work style of using a computer all day contributes to problems like obesity, hypertension (high blood pressure), diabetes and even increases the risk of heart disease. In using computers and related equipment, our bodies are often forced into uncomfortable postures for hours on end, and this can cause health problems for us.

Have you ever sat in front of your computer or game console for hours and finally left with a headache, back-ache, sore eyes or sore wrists/thumbs/arms? If you have, then you have already felt how computers can damage your health.

Repetitive Strain Injury (RSI) is a class of injuries (to tendons, etc.) caused by repetitive actions. They are commonly found in people using a computer for more than an average of 2 hours per day. The longer you work with computers / play games in this way, the worse the cumulative effect of the RSI can become – forcing people to seek medical treatment, surgery or even to have to change careers to give up a job that requires them to work with computers for long hours.

Carpal tunnel syndrome (CTS) and tendonitis are two of the many injuries that fall under the RSI-umbrella. CTS is caused by inflammation of the tendons in the wrist, causing pressure on the nerves in the wrist, resulting in pain and sometimes weakness in parts of the hand. This one of the most common types of injuries and many people require surgery to try to solve the problem.

Ergonomics is a science focused on trying to create a solution to this type of problem. It studies the human body and then tries to create designs for products so they better fit the way that we actually function. Its aim is to design products to fit us, so we won't have to force our bodies into unnatural and uncomfortable positions when using these products.

Ergonomics refers to the study and design of workplace for comfort and safety of people in work environments.

- For decades, very little progress was made to make the design of keyboards more ergonomic, and all of them came in a generic, square design. Nowadays, many keyboards feature a 'curved' or split design that promotes a more natural position for hands and wrists, with the palms turned slightly toward each other rather than flat relative to the keyboard.

- Ergonomic design is also an issue with mice as they are used so extensively in a GUI environment and as using a mouse requires users to hold their hands in unnatural, awkward positions for long periods of time.

Working in an environment which has been ergonomically designed needs to be coupled with sensible, healthy practices such as the following:

- Sit correctly – don't slouch or hunch.
- Adjust the height of your seat and/or table so you can sit with your hands at the keyboard while keeping your upper arms fairly slack at your sides, and your elbows and wrists level with each other. Your upper legs should be parallel to the floor and your feet flat on the floor.
- Make sure the lighting and ventilation/air circulation is adequate.
- Make sure that your mouse and keyboard are on the same level, and that the mouse is close to the keyboard.
- Make sure that your monitor is positioned at the correct level so that you are not straining your neck and shoulders to view the monitor.
- Every once in a while, turn your eyes away from your monitor and focus on something else which is far away. This will help prevent eye strain.
- Take regular breaks – don't work for more than an hour at a time without taking at least a small break.

The need for green Computing

Green computing refers to initiatives to design, use and disposal of technology in an environmentally or eco-friendly way.

Energy efficiency

One of the larger environmental issues being discussed globally is climate change. The argument is that the amount of carbon dioxide being added to the earth's atmosphere by human activity is responsible for causing large scale changes in global climate. The predictions are that, unless this is brought under control, it could have catastrophic consequences for the earth and for humanity.

How does this relate to ICT? Well, every computer needs electricity. Electricity needs to be created, and most of the world's electricity comes from burning fossil fuels like coal, gas or oil. Whenever you burn something, you add carbon dioxide to the atmosphere. So, the power required by computers is directly connected to the problem of climate change.

It may be hard to imagine that your single computer makes a difference – but what if your computer lab at school never had its computers switched off, and they used power even when no one was using them? What about office blocks full of computers that are not switched off? Can you imagine the power used by all the computers in the world (we are talking billions of computers here)?

When you start to think about the impact of computers on that level, it becomes easier to understand why improving energy efficiency in computers is very important in reducing the impact that computers have on our environment.

Design of hardware

Computer hardware designers have to try and balance producing 'faster' equipment with the increased use in power and the heat that these systems produce. In addition, there is a greater appreciation of the need to produce 'green' products that utilise less power. International standards such as Energy Star exist to promote the design and use of power-efficient hardware.

Power-saving settings

Many operating systems have options to effect power-saving by turning off or lowering the power to components such as monitors and hard drives after a pre-set period of inactivity. In addition, a system may hibernate, where most components are turned off. This is of course particularly important in a portable computer such as a laptop where one would like the battery power to be available for as long as possible.

Standby power

The electricity that many appliances waste just by being plugged in (even if they're switched off), is also known as standby power, or vampire power.

You switched the TV off with the remote control. You unplugged your cellphone from the charger as soon as it was charged. You switched your computer and your console off before you went to bed. You think that you are being responsible about your use of power. Right? Wrong! The problem is that many of the appliances that you think are 'off' are actually still very much 'on', and using power.

- The TV is not really off – it keeps using power so that it can respond to the remote.
- The cellphone charger is drawing power as long as it is plugged in and the wall socket is on.
- The same applies to your game console – and possibly to your laptop.

In fact, almost any place where you can see a little glowing LED light when a device is 'off', is a place where power is being sucked away like a vampire sucks blood in the night! The amount of electricity used by all our devices and their chargers even when they are off, can be surprising.

E-waste

Yearly we dispose of tons of e-waste. E-waste is the broken, old, outdated hardware that we don't want anymore. Most of it ends up in a landfill, often in third-world countries. Tons of cellphones alone hit our rubbish heaps every year, not to mention the other types of e-waste that we dispose of.

This is a serious environmental problem. Why? Because e-waste contains toxins. These toxins leak into the soil, creating problems for many years in the future when people want to use the land for other reasons, or when the toxins manage to reach and poison the water table (the water that ends up being taken from boreholes).

Computers and other technological devices contain amounts of heavy metals, such as cadmium, mercury, lead, etc.

It is very important for us to find ways of

- reducing toxins in technology manufacture
- forcing re-cycling of e-waste instead of letting it end up in landfills to poison future generations.

Paper

Hard copy (output on paper) is often seen as the wicked step-sister of output. Printers use paper. To make paper, you have to kill trees. The process of making paper also pollutes the planet with toxic chemicals, and it takes a lot of energy to make paper. All of this contributes to paper having a big carbon footprint.

The *paperless office* (where all documents are stored, transmitted and processed electronically) has long been the dream of fans of technology – if only for the sake of efficiency. Ultimately the need for printers will decrease, but we don't think that they'll be disappearing completely in the near future.

Think about these few examples:

- The reality is that small portable devices with high quality displays and long battery lives have only arrived recently to start to reduce the need for paper.
- Not everyone can afford the portable technology needed to reduce their dependence on paper – this is a huge part of the Digital Divide – poor, third world communities especially do not have access to this type of technology, whereas in wealthy, first world communities, everyone has access to computers or smartphones etc.
- Ultimately, a hard copy is the cheapest and easiest way to get copies of a finished product to the people who need to receive it (unless you know that they have the hardware and software needed to receive an electronic copy).
- It is certainly cheaper and less risky to string up a printed banner and paste up posters around town or hand out flyers than to try to set up and use large, visible video displays in public places.
- Hard copies are the only way to create non-electronic backups of data/information.

Printers will be around as long as there is any kind of need for a hard copy of the information produced by a computer system.

Tips on how to practice green computing

- Switch off computers and devices you are not using.
- Try and avoid the senseless wasting of paper by proofreading on the screen before you print. You can also save paper by printing '2 pages on 1' (most printers allow you to do this). Some printers allow for duplex or back-to-back printing.
- Rather than throw away hardware, see if someone else can make use of it or try to repair or upgrade it.
- Don't just throw away ink and toner cartridges. Rather send your cartridges back to the manufacturer when you have finished with them. The manufacturer will then be able to re-use them or dispose of them in an environmentally friendly way.
- Investigate the possibility of refilling and reusing ink and toner cartridges.

The impact of global e-communication

One of the ways in which computers have improved our world is by improving communication. You will cover more of this in the modules on e-communications and the Internet but it is important that you understand the significance of the way that computers have changed the way we communicate.

It has made communication:

- Faster
  - 100 years ago, sending a letter from Johannesburg to London could take 6 months. Now, with e-mail, it takes a few seconds.
  - You can talk to anyone, anywhere in the world (even if they are not in an office or even in a building) simply by pressing a few buttons on a phone.
  - You can do video chats and conferences with anyone, anywhere, at any time.
  - You can send huge files to anyone, anywhere, and know that they will receive perfect copies of these documents almost immediately.
  - Using services such as Twitter, you can get a message to millions of people instantly.
- Cheaper
  - Using technologies such as VoIP and programs such as Skype, you can communicate with someone at the other end of the world for hours, at a cost less than making a short long distance call using a normal telephone line.
  - Sending huge documents by e-mail is so cheap it is almost free!

- More effective and reliable
  - Digital document and data transfer means that the quality of what is received is identical to what is sent.
  - You can locate data/information almost instantly using the Internet.
  - You can do business transactions anywhere at any time.
  - At the click of a button you can see what has been going on in the lives of your friends.

ICT and e-communications has had a huge impact on our view of the world and our ability to communicate with the people in it.

1 Activity

1. Many jobs involve sitting in front of computers for long hours. Give three health problems associated with the lack of physical activity caused by sitting in front of a computer all day.

2. An advert for a new computer keyboard claims that it is ergonomically designed to reduce the risks of RSI and CTS.
   a) What does the term 'ergonomics' refer to?
   b) Give two possible features of a keyboard that is ergonomically designed.

3. Give three ways in which your work environment can be set up or designed to promote a healthy working environment, besides using ergonomically designed equipment.

4. Give two common-sense, healthy practices you should perform when working for long periods in front of a computer. This excludes any issues related to the physical environment or ergonomic issues.

5. Everything is going green – even computing! What does the term 'green computing' refer to?

6. E-waste has become a major concern in developing countries. Explain what this means by explaining what e-waste is.

7. The 'paperless office' has not lived up to its initial promises. The rapid destruction of forests to make paper is a now a global environmental crisis.
   a) What do we mean by the paperless office?
   b) Give three reasons why our use of hard copies has not decreased, as originally thought would be the case.
   c) Give three practical ways in which the amount of paper used for printing can be reduced.

8. Give two ways to contribute to green computing other than saving paper.

9. Give three ways in which e-communication has changed the way we communicate.

Glossary

Carpal Tunnel Syndrome	This is caused by inflammation of the tendons in the wrist, causing pressure on the nerves in the wrist, resulting in pain and sometimes weakness in parts of the hand.
E-waste	The broken, old, outdated and discarded hardware that we don't want and don't need.
Ergonomics	Studies of the human body to try to design products so they better fit the way that we actually function.
Green computing	Green computing refers to initiatives to design, use and disposal of technology in an environmentally or eco-friendly way.
RSI (Repetitive Strain Injury)	A class of injuries (to tendons, etc.) caused by repetitive actions.

MODULE 5.3

Safe Internet and e-mail use

Introduction

The Internet and its applications are wonderful tools for communicating and doing research. However, there are a number of hazards and potentially harmful threats that you need to know about. The more you know about these issues, the better equipped you are to be able to carry on working on the Net without disruptions to your work or your computer.

Let's look at a few of these, starting with the biggest threat, computer viruses.

Malware

In recent times the term *malware* (**mal**icious **ware**) has arisen. It is used to describe any software that is written with negative intentions. These can range from highly destructive software, to software that is more of annoyance than anything else.

Viruses

A computer virus is software that is written to disrupt the normal functioning of a computer and is loaded onto a computer without the knowledge or consent of the owner. Many viruses are also designed to create and exploit security loopholes in systems.

They are written in such a way as to try to reproduce or replicate themselves and spread to other computers. These programs often corrupt files including system files of the operating system, making your computer or other hardware devices unusable. Note that they do not attack the hardware but rather the software that manages and controls the hardware.

How does a virus spread?

Viruses can be spread via

- infected attachments sent with e-mail messages
- infected downloaded files from the Internet
- infected files stored on portable media such as a flash drive.

Most viruses activate when you run or open the infected file.

How do I know whether my computer has a virus or not?

It is important to know that not every computer problem is caused by a virus. Their effects vary but some of the tell-tale signs are:

- Files suddenly start disappearing.
- The computer slows down dramatically.
- Programs stop working properly.
- Free space on the computer suddenly gets greatly reduced.
- Error messages that you have not seen before, start appearing.
- The computer no longer boots up.
- Your anti-virus software reports the threat or presence of a virus.

Virus prevention

The two main measures you can take to prevent your computer from being affected by viruses, are to make use of anti-virus software and to act responsibly and use common sense when working on a computer and the Internet.

Anti-virus software

Anti-virus software is a computer program that scans computer files for viruses and eliminates them. It detects the virus by scanning the computer file and comparing it to its database of virus definitions.

A virus signature or definition is the pattern that uniquely identifies a virus. These definitions need to be updated often and can be downloaded from the Internet for the anti-virus software you have installed.

An anti-virus program usually runs in the background and is usually set to start automatically when the computer is switched on. It can be configured to

- warn you of the presence of a virus or any suspicious activity on your computer
- automatically delete any virus it detects
- quarantine a virus. This simply means the virus is placed in an area of the disk where it cannot infect the computer.

Examples of anti-virus software include Norton's Antivirus, AVG, Avast, Bit Defender, Kaspersky and McAfee. Some of these are free to download and can be used for a limited period of time as shareware.

General guidelines

You can play your part in preventing viruses by

- making sure you have an anti-virus software package installed
- keeping your virus definitions (signatures) regularly updated as new viruses come out every day
- setting your anti-virus program to update automatically when you are connected to the Internet
- scanning files stored on portable media, such as flash drives before using them
- scanning e-mail attachments before opening them
- setting your anti-virus software to scan incoming and outgoing e-mail
- not opening e-mail or running programs from people you do not know
- avoid pirated software because malware is often packaged with pirated software
- disabling the autorun feature for flash drives.

Flash disks and viruses

Viruses are increasingly being spread via infected flash disks. This is one of the reasons why some companies ban their use.

Flash disks can automatically run certain commands when they are inserted in a PC. This so-called autorun feature is a prime source of activating and spreading viruses.

Computer worms

A computer worm is malware that is able to distribute itself over a network, normally via e-mail, without a person having run an infected program. It is able to reproduce itself many times and so your computer could send out hundreds of these worms with devastating effect. One example is a worm that is able to send a copy of itself to everyone in your e-mail address book, without you knowing, potentially infecting the computers of all those people and sending itself out again to all the people in the receiver's address book.

Worms can also reproduce so many times that they take up your entire computer's available memory or free hard disk space. This causes your computer to run very slowly and possibly even crash (stop working).

Computer worms can usually be prevented by having updated ant-virus software installed on your computer.

Trojans

A Trojan is a destructive program disguised as an useful application. You are tricked into opening it as it seems to be legitimate software, so you open and run it. You may for example receive an e-mail attachment and when you open it, it activates the Trojan. Most e-mail systems will not allow you to send exe files as attachments.

This program could delete files on your computer or even scan your computer for personal information and send it off to the person who created it. Unlike viruses and worms, Trojans do not reproduce themselves by infecting other files. They rely on you running the program, so don't open attachments if you're not completely sure it is safe to do so.

Spyware

Spyware is software that tries to monitor and track the way you use your computer. It then sends this information off to third parties – without your consent or knowledge. Spyware is installed on a computer without the user's knowledge.

The most dangerous forms of these applications are called keyloggers. They capture your keystrokes as you type (and possibly even the movements of your mouse and screenshots of what is being displayed) in an attempt to find out what your usernames and passwords are.

Signs that a computer could be infected by spyware

- It is reported by anti-spyware software.
- There could be Internet activity not initiated by the user.
- Changes are made to Internet settings, such as the home page, but not by the user.

- The computer slows down dramatically.
- The user is redirected to different pages when trying to go to specific pages on the Web.
- Many pop-up windows appear, often with advertising or requests to install software.

Prevention

- Ensure that you have anti-spyware software installed. This is often part of your anti-virus package or security software.
- Anti-spyware needs to be updated to update a list of known spyware, as well as Internet sites known to infect computers with spyware.
- Ignore pop-up windows.

Adware

Adware is software that downloads and bombards you with adverts that pop-up whenever you are connected to the Internet. It is more of a nuisance than a threat but anti-adware software prevents this software from working on your computer. Often anti-spyware software is combined with anti-adware software in a single package. Most web browsers provide options to block pop-ups but some websites will not function properly if the pop-ups are blocked.

Spam

Spam is the electronic equivalent of 'junk mail'. It is an e-mail being sent to you in the form of advertising products that you did not request. Spam has become a huge worldwide problem. The main problem is that it takes time to sort out and delete the unwanted e-mails, which also take up unnecessary space.

Prevention tips for spam

- Check if your ISP can help. Many service providers filter out as much of the spam as possible.
- Be careful of who you give your personal details and e-mail address to. Many websites and even some organisations 'sell' lists of contact details.
- Investigate the capabilities of your e-mail program. Most programs have the ability to set up rules to filter out spam and automatically send it to a so-called Junk or Spam folder. These rules can be based on words appearing in the content of the e-mail or if e-mail comes from a particular sender.

- Most anti-virus software packages have anti-spam software included. Specialised anti-spam software also exists.
- Never respond to spam (this merely confirms a valid e-mail address).

Hoaxes and scams

Internet users often do not recognise whether an e-mail message is a hoax. The Internet community has been inundated by e-mail hoaxes and pranks over the last few years.

An Internet hoax is an e-mail message or article intended to deceive or defraud others.

False information is not always meant to be malicious but be aware of e-mail message that are meant to threaten or cheat you. E-mail scams can include get-rich-quick schemes, fad diets and threatening chain letters.

Many hoaxes encourage people to send the message to as many recipients as possible so that they can get a reward for themselves or on behalf of someone else. Many of these e-mails are used to gather e-mail addresses for advertising (spam) purposes.

How to protect yourself from hoaxes and scams

- Never accept information at face value. Always question what you read on the Internet.
- Always check the source of the information and try to determine whether it is reliable. E-mail without a source or author is most probably unreliable. Don't be fooled by documents with official looking letterheads – these can be easily forged.
- When people try to legitimise the information by saying that it happened to a friend of a friend, this is often the basis of an urban legend and not based on fact.
- If you recognise a message as being a hoax, delete it and do not forward it. Send a message back to the sender warning them about the hoax.
- Be particularly aware of virus hoaxes that encourage you to delete files that may actually be crucial to your computer functioning correctly.

There are many sites on the Internet that investigate hoaxes and publish their results. You can go onto these sites and check the validity of the message.

Activity

A hoax is a deliberate attempt to deceive or trick people into believing or accepting something which the hoaxer (the person or group creating the hoax) knows is false. Many reports of hoaxes can be found on the Internet. Find such a hoax and provide feedback on the way and the reason the hoax was created.

Identity theft is when someone steals your personal details such as ID number, credit card or driver's licence details to commit fraud or another type of crime electronically. For instance, that person can apply for jobs and loans or purchase articles using your stolen details. You can suffer severe consequences, as you could be held accountable for the actions of the person who stole your identity.

Phishing

Phishing refers to attempts from people to con the user into giving out personal and confidential details such as PIN numbers and passwords by posing as a legitimate organisation, often a bank, usually via e-mail.

In the e-mail they request that you give personal information such as your password and PIN as if they are 'checking' the information.

In addition, the e-mail often requests that you visit a website (banking or shopping) by clicking on a link, to update this information. The website that you visit has been set up to look like the official website, but it is actually a fake website. As soon as you enter your information, they can use it to access your accounts and steal your money or make purchases on your behalf.

Prevention tips for phishing

- Do not respond to requests via e-mail, from any source, to confirm details such as bank account numbers and especially PINs and passwords.
- Type in the URL of bank websites etc. directly and do not click on a link in an email to go to these sites.
- Check that the site is secure.

Secure URL

When a URL begins with https:// or a small closed lock is displayed somewhere on the page, often next to the URL, it means that the site is encrypted for security.

Pharming

Pharming is similar to phishing but the user does not receive an e-mail. Pharming is an identity theft scam where the user's computer is infiltrated so that they are automatically redirected to another (fake) website even if they type in the correct URL (address) for that website. The criminal is then able to use any data the user enters. Pharming is very hard to detect and prevent.

Prevention tips for pharming

- Ensure that you have up to date anti-spyware software installed.
- Many sites requiring secure information are now created so that browsers will warn you if you are re-directed to another site. Always pay attention to these warnings.
- Ensure that the site you are accessing is secure.

E-mail spoofing

E-mail spoofing is the changing of an e-mail header so that the origin of the e-mail appears to be from a different source. It is used in spam and phishing e-mails to disguise the origin of the e-mail message. For example, a spoofed e-mail may look like it is from someone in a position of authority, making the e-mail more credible.

2 Activity

Choose a term/concept from Column B that matches the description in Column A. Write only the letter next to the question number.

	Column A	Column B	
1	Any software that is written with negative intentions.	A	Trojan
2	A type of program that is loaded secretly on a PC to disrupt the functioning of the PC. They also try to reproduce or spread.	B	Hoax
3	Software designed to distribute itself over a network and reproduce so many times that your computer's available memory or free hard disk space is used up.	C	Pharming
4	A destructive program disguised as a useful application. You are tricked into opening it as it seems to be legitimate software, so you open and run it.	D	Phishing
5	An e-mail message or article intended to deceive or defraud others.	E	Malware
6	The electronic equivalent of 'junk mail'.	F	Worm
7	Attempts to con the user into giving out personal and confidential details by posing as a legitimate organisation, often a bank, usually via e-mail.	G	Spam
8	A situation where the user's computer is infiltrated so that they are taken to another (fake) website even if they type in the correct URL (address) for the site.	H	Virus

Guidelines using the Internet and e-mail with Courtesy and safety

Besides some of the specific issues already discussed, a number of basic, common-sense guidelines exist to help you to when it comes to using the Internet with courtesy and safety.

Netiquette

Netiquette (Internet + Etiquette), is the name given to the use of good manners and showing respect for other users when using the Internet.

Some guidelines include the following:

- Ensure that any e-mail message you type is clear and contains no errors.
- Never type e-mail messages only in capital letters. If you do, it shows that you are 'shouting' at the other person.
- Always ensure that any file that you attach to an e-mail message is free of viruses.
- A smiley or emoticon (short for emotion icon) can be used to express emotions in messages. They are also found in SMS language. Emoticons should only be included in informal correspondence, or in messages to people you know personally. Including them in an e-mail to your employer or to someone you only know in a professional capacity can seem familiar or rude. Below are a few examples. To view them, turn the page sideways.

:-)	Smile	:-D	Laugh	8-)	Sunglasses
:-(	Sad	:-x	Small kiss	(((h)))	Hug
;-)	Joking / winking	:-X	Big kiss	:-0	Bored

- Never be insulting in a message. If you wish to tease someone, use a smiley.
- Never pass on chain mail, hoaxes or spam. It just causes unnecessary traffic on the Internet and can irritate people.
- Make sure that any files you attach to a message are not too large as this takes too long to download and can cost the recipient money if it uses up their cap.
- If you are forwarding an e-mail message to many people, rather add their addresses in the Bcc field. In this way the message will not be seen as spam and you are not giving out people's e-mail addresses to many people.
- Be careful of passing on too much personal information in an e-mail message, as someone may intercept it.

General guidelines

- Keep your software up to date.
- It is always a good idea to keep your operating system up to date. Most of these programs allow you to have the updates automatically sent to you when you are connected to the Internet. Keeping the software up to date can prevent security problems.
- Keep your anti-virus software's definitions up to date.
- This prevents your computer from being infected by new types of viruses and worms which come out almost on a daily basis.
- Be careful when installing software from the Internet.
- Be careful when installing free software from the Internet, even if it is a free background or extra cursors, as malware is often 'bundled' with these programs. To be safe, never install any software when asked to, unless you know what it is you are installing. Some examples of software you can install safely is well-known software such as WinZip, Flash, Adobe Reader and QuickTime.
- Do not click on links in pop-ups.
- Be aware of pop-up windows that offer you a prize, offer to speed up your computer or display 'Windows error messages' with solutions to correct them, as these will direct you to another site to install malware. Close this type of window immediately and make sure that you do not click on any links within the pop-up window. Most web browsers and anti-virus software now provide some protection against pop-ups.
- Don't open attachments or programs unless if you are sure of what they are.
- Many virus infections are caused by users unsuspectingly opening attachments in e-mails or running and installing programs where they are not sure of their origins.
- Have good password policies.
- Don't give out personal information such as identity numbers unless you are absolutely sure it is necessary to do so.
- Stay informed – keep abreast of reports in papers and tips supplied on banking websites..
- Don't be gullible – if something sounds too good to be true, it probably is.

Safety when interacting with others online

Chat rooms, instant messaging and social networking sites such as Facebook are all the range. You need to be very careful how you react and behave online.

- You always need to be aware that the person with whom you are communicating may not be who they say they are.
- Never be tempted to give out any personal details such as your name, address or telephone numbers to people you meet online.
- Do not make arrangements to meet someone you meet online. If someone wants to meet you in person, inform an adult and do not make any arrangements to meet them.
- If someone makes you feel uncomfortable by asking personal questions or using bad language, then discontinue the 'conversation' immediately and tell your parents.
- Don't be tempted to use your real name in chat rooms. Rather use a nickname or alias that does not reflect your gender or age.
- Never accept file transfers from strangers. Some chat programs allow users to exchange files. These files can potentially contain viruses or other harmful items. If possible, turn off this feature in your chat software.
- Think before you post photos or text anywhere on the Internet. Photos, comments, and blog entries can be saved and forwarded to other people. You should only post photos and text that you wouldn't mind other people, including your parents, to see.
- Be aware that no page you create on social networking sites is private. Once you have posted a comment or photo, it is in the public domain. Ensure that you understand and use the privacy and security settings that social networking sites provide so that you can specify who can access the information you post.
- Don't post embarrassing photos or comments on your 'friends' social networking pages.
- It is acceptable to ignore friend requests from strangers on social networking sites.
- Be suspicious of someone who is just too 'perfect'. The details of what you like and enjoy are available for everyone to see. It is therefore very easy for someone to take that information and fake being your 'soul mate' to gain your trust. (This is how paedophiles work.)

Activity

1. You received the following e-mail, supposedly sent by an official at General bank. Examine the e-mail carefully and answer the questions below:

From: General Bank [mailto:mary@general.co.za
Sent: 17 May 2011 02:38 PM
To: jack@domain.co.za
Subject: General Bank E-mail Verification –jack@domain.co.za

Dear valued General Bank Member,

This e-mail was sent by the General Bank server to verify your e-mail address. You must complete this process by clicking on the link below and entering in the small window your General Bank User ID and Password

This is done for your protection – because some of our members no longer have access to their e-mail addresses and we must verify it. To verify your e-mail address and access your bank account, click on the link below:

http://www.general.co.za/wjwwU3gcnUhkTrqcR9AmuEvaPKkmvqsegOptMRPAqYof9UecGDV0xoNa3f0s3cz0a2

a) What type of scam is this?

b) Should a person click on the link? Motivate your answer.

2. Explain the difference between spyware and a computer virus. Give a solution for preventing becoming a victim of each of these threats.

3. Computer viruses are a growing problem and cause many problems on computers daily.

a) What is a computer virus?

b) Discuss two ways in which a computer can become infected with a virus.

c) Why does one need to update anti-virus software?

d) Give the name of one popular anti-virus software package.

e) How does a computer worm differ from a virus?

f) How does a Trojan differ from a virus?

4. Another Internet-related problem is spam.

a) Briefly explain what spam is.

b) Give two ways in which you can try to prevent spam.

5. Chat rooms are experiencing a renewed interest on the Web. Give three tips to help ensure one's safety in chat rooms.

6. The following is an e-mail that has been sent:

To...	thandi.khumalo@hullbibliophiles.org.uk
Cc...	bookworm@gmail.com
Bcc...	
Subject:	
Attach:	Our Library (18.5 GB)

Hi

SORRY I HAVE NOT CONTACTED YOU FOR A WHILE BUT I HAVE BEEN VERY BUSY AT WORK. WE HAVE BEEN WORKING ON A PROJECT CALLED 'COMPUTERS PART OF YOUR LIFE'.IT IS A COMMINITY BASED COMPUTER LITERACY PROJECT AND IS VERY EXCITING.

You won't recognise the library since you were here last, so I attached a photo of it after the renovations.

Love
Ilse

BTW My boss got a huge bonus and is apparently going to use it for plastic surgery on her nose :-0

a) Define the term netiquette.

b) Give three netiquette rules that have been ignored in the e-mail above

7. Your younger sister has joined Facebook.

a) What are some of the dangers she might be exposed to by using a social networking site?

b) Give her some common sense guidelines to help her protect herself while using a site such as Facebook.

8. Modern e-communication tools such as social networking sites and IM are often associated with identity theft.
   a) Describe what is meant by the term identity theft.
   b) Explain why these tools are often associated with identity theft.

9. Spyware is becoming a huge problem for computer users.
   a) Briefly describe what spyware is.
   b) Discuss two signs that your computer could be infected with spyware.

Glossary

Adware	Adware is software that downloads and bombards your computer with adverts that pop-up whenever you are connected to the Internet. It is more of a nuisance than a threat.
Anti-virus software	Anti-virus software is a computer program that scans computer files for viruses and eliminates them.
Computer virus	A computer virus is a program that is written to disrupt the normal functioning of a person's computer without their knowledge or consent.
Computer worm	A computer worm is malware that is able to distribute itself over a network, normally via e-mail, without a person having run an infected program.
E-mail spoofing	E-mail spoofing is the changing of an e-mail header so that the origin of the e-mail appears to be from a different source.
Identity theft	Identity theft is when someone steals your personal details such as your ID number, credit card details or driver's licence to commit fraud or another crime.
Internet hoax	An Internet hoax is an e-mail message or article intended to deceive or defraud others.
Netiquette	Netiquette (Internet + Etiquette), is the name given to the use of good manners and showing respect for other users when using the Internet.
Pharming	Pharming is an identity theft scam, where the user's computer is infiltrated so that they are automatically redirected to another (fake) website, even if they type in the correct URL for that website.
Phishing	Phishing refers to attempts from people to con the user into giving out personal, confidential details such as PIN numbers and passwords, by posing as a legitimate organisation, usually via e-mail.
Secure URL	A secure URL begins with https:// or has a small closed lock displayed somewhere on the page, often next to the URL.
Spam	Spam is the electronic equivalent of 'junk mail'. It is an e-mail being sent to you in the form of advertising of products that you did not request.
Spyware	Spyware is software that tries to monitor and track the way you use your computer. It is installed on users' computers without their knowledge.
Trojan	A Trojan is a destructive program disguised as an useful application.
Virus signature (definition)	A virus signature or definition is the pattern that uniquely identifies a virus.

Connectors on a Computer

Communications

RJ 45 (Ethernet) cable connection

802.11 x WiFi connection

WiFi is built into devices such as laptops, tablets and even cellphones - you don't usually see a connection only the WiFi symbol shown above.

Sometimes Bluetooth is used - that's the symbol shown here below.

Sound

Computers tend to have two small jack sockets such as the one shown here. They accommodate a microphone and a headphone / external speakers. The use of the jack is usually shown by the following symbols:

Video

VGA (Video Graphics Array)

DVI (Digital Video Interface)

The connectors below are used to connect your computer to a TV instead of a computer monitor. You are most likely to find one such connector on a laptop / tablet.

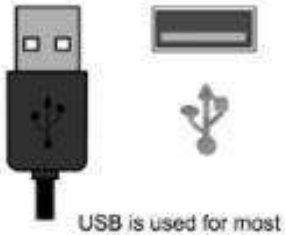
S Video

Composite Video

HDMI (High Definition Media Interface)

General Peripherals

USB (Universal Serial Bus)



FireWire (IEE 1394)

USB is used for most peripherals whilst Firewire tends to be used for specialised video / external HDDs. Both have a smaller connector option as well as the one shown here. The symbols shown are used to identify the ports.

Other connections

Other connections can be visible on a computer (especially older models). They include:

- Parallel ports (used to be used mainly to connect printers)
- Serial (RS 232) ports (modem and a mouse)
- PS2 connectors (keyboard and mouse)
- Keyboard connector (large connector for a keyboard)
- SCSI port (for printers, scanners and even HDD)

These connections (or ports as they can be called) have all been replaced by the USB port which most peripherals now use to connect to the computer. Even Firewire is used only in a limited fashion by professional video editors / musicians.