

2.2 Programming fundamentals	Data types	Integer	"A data type used to store positive and negative whole numbers."
	Data types	Real	"A data type used to store an approximation of a real number in a way that can support a trade-off between range and precision. A number is, in general, represented approximately to a fixed number of significant digits and scaled using an exponent."
		Boolean	"Used to store the logical conditions TRUE / FALSE. Often translated to On/Off, Yes/No etc."
		Character	"A single alphanumeric character or symbol."
		String	"A sequence of alphanumeric characters and or symbols. e.g. a word or sentence."
		Casting	"Converting a variable from one data type to another. e.g. variable entered as a string, but needs to be an integer for calculation. age = INPUT("Enter your age: ") age = INT(age)"
	Additional programming techniques	String manipulation	"Commands and techniques which allow you to alter and extract information from textual strings e.g. .length .substring(x, i) .left(i) .right(i) .upper .lower ASC(...) CHR(...)"
		File handling: Open	"File handling is the process of dealing with input to and from files. Files first have to be opened, this creates a handle to the file and allows reading and writing."
		File handling: Read	"File handling is the process of dealing with input to and from files. Once a file has been opened it is possible to use commands to read its contents and return them to your program."
		File handling: Write	"File handling is the process of dealing with input to and from files. Once a file has been opened it is possible to use commands to write data to file from your program."
		File handling: Close	"File handling is the process of dealing with input to and from files. Once you are done reading / writing it is important to close a file, this releases the file handle and breaks the connection between it and your program."
		Record	"A data structure which consists of a collection of elements, typically in fixed number and sequence and typically indexed by names. The elements of records may also be called fields."  "The record type is a data type that describes such values and variables. Most modern computer languages allow the programmer to define new record types. The definition includes specifying the data type of each field and an identifier by which it can be accessed."
		SQL	"The language and syntax used to write and run database queries"





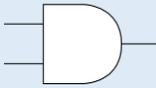
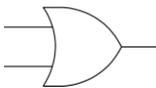
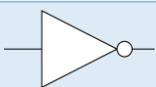
		SQL command: SELECT	<p>“A key word in the SQL programming language used for the querying (retrieval) of data.” e.g.</p> <pre>SELECT Name, Age, Class FROM Students_table WHERE Gender = 'Male'</pre>
		SQL command: FROM	<p>“A key word in the SQL programming language used to signify which table(s) we are using.” e.g.</p> <pre>SELECT Name, Age, Class FROM Students_table WHERE Gender = 'Male'</pre>
		SQL command: WHERE	<p>“A key word in the SQL programming language used to filter the results of your query.” e.g.</p> <pre>SELECT Name, Age, Class FROM Students_table WHERE Gender = 'Male'</pre>
		Array	“A set of data items of the same type grouped together using a single identifier. Each of the data items is addressed by the variable name and a subscript.”
		Sub programs	“A block of code given a unique identifiable name within a program. Supports code reuse and good programming technique.”
		Procedure	“A block of code given a unique identifiable name within a program. A procedure can take either zero or more parameters when it is called. The procedure should be designed and written to perform one task or action which is clearly indicated by its name.”
		Function	“A block of code given a unique identifiable name within a program. A function can take either zero or more parameters when it is called and should return a value. The function should be designed and written to perform one task or action which is clearly indicated by its name.”
		Random number generation	“Most programming languages have built in functions or libraries that allow you to easily generate random numbers. Creating truly random numbers is actually something quite difficult for a computer, and these algorithms are quite complex.”
2.3 Producing robust programs	Defensive design	Defensive design	“Defensive design is the practice of planning for contingencies in the design stage of a project or undertaking.”





		Anticipating misuse	“The ability of a programmer to consider how the end user might accidentally (or on purpose) break the program and then to write additional code to handle these situations.”
		Authentication	“Verifying a user identity before they can use a program with username and password. Strong passwords over a certain length with symbols and mixed case are advised.”
		Input validation	“Ensuring data input by the user meets specific criteria before processing. Range check. E.g. between 1 and 31. Type check. E.g. number not symbol. Presence check. E.g. data has been input. Format check. E.g. postcode is LLN(N) NLL.”
		Maintainability	“A selection of techniques and methods that make code easy to debug, update and maintain.”
		Naming conventions	<p>“Many programmers / organisations use certain naming conventions for their variables / contents / procedure names etc.</p> <p>Camel case is a popular one used in industry where the first word of an identifier uses all lower case, with all subsequent words starting with a capital letter:</p> <p>e.g. studentsFirstName”</p>
		Indentation	“Indenting makes it easy to see where structures begin and end. Conditions and iterations should be indented. Code inside procedures and functions should be indented.”
		Commenting	“Used by a programmer to explain sections of code. Ignored by the compiler.”
	Testing	Testing	“This involves testing the program under various conditions to make sure it is going to work. You need to think about what devices it could be used on and what might cause the program to crash.”
		Iterative testing	“Each module of a program is tested as it is developed.”
		Final/terminal testing	“Testing that all the modules of a program work together as expected. Checking the program meets the expectations of the user with real data.”
		Syntax error	“Rules of the language have been broken. The program will not run. Variables not being declared before use. Incompatibility of variable types. E.g. sum = A Using assignments incorrectly. E.g. 2 + 2 = x Keywords misspelt. E.g. PRNT(“Hello”)”
		Logical error	“The program runs but does not give the expected output. Division by zero. Infinite loop. Memory full. File not found.”
		Test data	“Values used to test a program, includes normal test data, boundary test data and erroneous test data.”



		Test data: Normal	<p>“Data supplied to a program which you would expect.</p> <p>e.g. A program has been written to average out test scores from students, the scores allowed are from 0-100. Normal test data could be: 32, 40, 82 etc.”</p>
		Test data: Boundary	<p>“Data supplied to a program which is designed to test the boundaries of a problem.</p> <p>e.g. A program has been written to average out test scores from students, the scores allowed are from 0-100. Boundary test data could be: -1,0,1 or 99,100,101”</p>
		Test data: Invalid	<p>“Data of the correct type but outside accepted validation limits.</p> <p>e.g. a program asks for the user to input a whole number from 0-100 then examples of invalid data could be -5, 150 etc.”</p>
		Test data: Erroneous	<p>“Data of the incorrect type which should be rejected by a computer system.</p> <p>e.g. a program asks for the user to input a whole number from 0-100 then examples of erroneous data could be the string ‘hello’ or the real 3.725 etc.”</p>
<b>2.4 Boolean logic</b>	Boolean logic	Logic diagram	<p>“A method of expression Boolean Logic in a diagrammatic form using a set of standard symbols representing the various Logic Gates such as AND NOT OR NAND etc.”</p>
		Logic gate	<p>“An individual symbol used in a logic diagram which represents a single gate e.g. AND, OR, NOT.”</p>
		Logic gate: AND	 <p>“A logic gate which accepts two inputs and produces one output. Both inputs must be TRUE (1) for the output to be TRUE (1), otherwise the output is FALSE (0).”</p>
		Logic gate: OR	 <p>“A logic gate which accepts two inputs and produces one output. At least one input must be TRUE (1) for the output to be TRUE (1), otherwise the output is FALSE (0).”</p>
		Logic gate: NOT	 <p>“A logic gate which accepts one input and produces one output. If the input is TRUE (1) then the output will be FALSE (0). If the input is FALSE (0) then the output will be TRUE (1).”</p>
		Truth table	<p>“A notation used in Boolean algebra for defining the output of a logic gate or logic circuit for all possible combinations of inputs.”</p>
<b>2.5 Programming languages and IDEs</b>	Languages	High-level language	<p>“A language designed to help a programmer express a computer program in a way that reflects the problem that is being solved, rather than the details of how the computer will produce the solution. One-to-many language.”</p>





		Low-level language	"A language which is close to machine code. Related closely to the design of the machine. A one-to-one language."
		Translator	"A program that translates a program written in assembly language into machine code."
		Compiler	"A program that translates a high-level language program, source code, into a computer's machine code."
		Interpreter	"Translates and executes a program one statement at a time."
	The Integrated Development Environment	IDE	<b>Integrated Develop Environment:</b> "A software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger."
		IDE: Error diagnostics	"These are tools provided by IDE's which give detailed feedback on errors in your code."
		IDE: Run-time environment	"A configuration of hardware and software. It includes the CPU type, operating system and any runtime engines or system software required by a particular category of applications."

