Notes that Excel

A very compressed computing textbook. Content follows the syllabus (7155) document. Made by Elvberth Yobel Rondonuwu

Module 1: Computing Fundamentals

1.1 Computer Architecture

[**Computer**] - Device that receives and processes data according to a set of instructions and produces the processed data
[**Computer architecture**] - Description of how a computer is designed and built to function, including how its various parts are designed, organised and connected
[**Data** (singular: datum)] - Information used in a computer program

1.1.1 Perform calculations using bits, bytes, kilobytes, kibibytes, megabytes, mebibytes, gigabytes, gibibytes, terabytes, tebibytes, petabytes and pebibytes.

[**Bit**] - Binary digit, value of either 0 or 1
[**Byte**] - Binary number made of 8 *bits*

'Bi's - powers of 1024
Normal units - powers of 1000
>> *Power of 1 starts from 'ki'*

Bits → *Bytes* → *Kilo* → *Mega* → *Giga* → *Tera* → *Peta* (just rmb your own data storages)
Bits → *Bytes* → *Kibi* → *Mebi* → *Gibi* → *Tebi* → *Pebi*

1.1.2 Describe the function of key components of a computer system

[**Processor**] - Processes data and follows instructions
>> [**Execute**] - To follow or perform an instruction
>> [**Software**] - A set of instructions to perform specific tasks on a computer
[**Main Memory**] - Stores data and instructions temporarily for immediate use by the processor
>> [**Address**] - Number used to locate a byte in memory
>> [**Volatile**] - Data is lost when power supply is interrupted
[**Secondary Storage**] - Stores large amounts of data that will not be lost when the power supply is interrupted
>> [**Hard disk/drive**] - Secondary storage where data is stored on rigid rotating disks coated with magnetic material

1.1.3 Describe the function of data and address buses in reading and writing to memory

[**Data Bus**] - Transports data between memory and processor; bi-directional
>> [**Bi-directional**] - Able to work in two directions, to and fro
[**Address Bus**]- Transports required memory location from processor to memory; uni-directional
>> [**Uni-directional**] - Able to work in one direction only

> When *writing* data, data + address bus used to send data and specify location from processor.
> When *reading* data, address bus specifies location from processor, then data sent back through data bus to processor.

1.1.4 Describe different input/output interfaces in terms of typical applications, connectors and speed

[**Input**] - Data or instructions that the computer receives for processing
[**Output**] - Intermediate or final results produced by the computer; usually in the form or processed data
[**Hardware**] - Physical components of computer
[**Input device**] - Hardware device that allows user to enter data and instructions into a computer
[**Output device**] - Hardware device used to display, project or print processed data from a computer
[**Lane, PCI Express**] - Interface for transferring data between a computer and an expansion card
[**Motherboard**] - Main circuit board in computer that connects all the components together

Input/output devices	**Typical applications**	**Connectors**	**Maximum speed**

High-Definition Multimedia Interface (HDMI)	- Delivering audio/video data to compatible devices	- HDMI Standard
		- HDMI Mini
		- HDMI Micro
			- HDMI 1.3-1.4b
			  - 10.2 Gbit/s
			- HDMI 2.0-2.0b
			  - 18 Gbit/s
			- HDMI 2.1
			  - 48 Gbit/s

Universal Serial Bus (USB)	- Powering and/or communicating with external devices	- USB Type A
		- USB Type B
		- USB Mini
		- USB Micro
		- USB Type C
			- USB 2.0
			  - 480 Mbit/s
			- USB 3.2
			  - 20 Gbit/s
			- USB 4
			  - 80 Gbit/s

Peripheral Component Interconnect Express (PCI Express)	- Communicating with internal expansion cards (e.g. graphics	- PCI-e x1 - PCI-e x4 - PCI-e x8	[Increases with number of lanes, i.e. up to 16]

	cards)	- PCI-e x16	Max speed per lane:
			- PCI Express 5.0
			  - 4 GB/s
			- PCI Express 6.0
			  - 8 GB/s
			- PCI Express 7.0
			  - 16 GB/s

1.1.5 Describe the use of magnetic, optical and solid-state media for secondary storage in terms of durability, portability, typical capabilities, cost and speed

	Magnetic	**Optical**	**Solid-State**
Description	- Data stored on magnetic material, read/written by magnetic 'head'	- Data stored as small pits or indentations, read/written by laser	- Data stored in electronic circuits, no moving parts
Durability	Most vulnerable - Magnetic fields - Heat - Impact - Natural deterioration over time	Vulnerable - Scratches - Natural deterioration over time More resistant to - Heat - Impact	Most durable - Impact - Temperature changes
Portability	Heaviest and bulkiest	Portable due to small size and lightweight	
Typical Capacities	Up to TBs	Up to GBs	Up to TBs
Cost per GB	Lowest	Mid	Highest
Speed	Slower than SS		Fastest

1.2 Data Representation

1.2.1 Represent positive whole numbers in binary form

[**Denary number system (or decimal number system)**] - A number system that is made up of 10 unique digits

[**Leading zero**] - The zero digit (0) that is to the left of the first non-zero digit in a number

[**Binary number system**] - A number system that is made up of two unique digits
[**Hexadecimal number system**] - A number system that is made up of 16 unique digits

1.2.2 Convert positive whole numbers from one number system to another - binary, denary and hexadecimal; and describe the technique used

[Binary to Denary]

To convert a binary number to its denary equivalent, we use the sum of place values method, i.e., sum the product of each digit and its place value. Let us use the binary number 11 1001_2 as an example.

Place value	256	128	64	32	16	8	4	2	1
	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Binary digit				1	1	1	0	0	1

Table 2.4 11 1001_2 in the binary number system

$$(1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$
$$= 32 + 16 + 8 + 0 + 0 + 1 = 57$$

Figure 2.7 Converting 11 1001_2 to denary

Hence, the denary equivalent of 11 1001_2 is 57.

[Hexadecial to Binary]

To convert a hexadecimal number to its binary equivalent, we map each hexadecimal digit to its four-digit binary equivalent as shown in Table 2.16.

Hexadecimal digit	0	1	2	3	4	5	6	7
Binary value	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal digit	8	9	A	B	C	D	E	F
Binary value	1000	1001	1010	1011	1100	1101	1110	1111

Table 2.16 Hexadecimal digits and their four-digit binary equivalents

For example, let us convert $3C_{16}$ to its binary equivalent.

1. Write down $3C_{16}$ as its individual hexadecimal digits, i.e., '3' and 'C'.
2. In Table 2.17, we write down the four-digit binary equivalents of '3' and 'C' found in Table 2.16, i.e. '0011' and '1100' respectively.
3. Finally, the sets of binary digits in Table 2.17 are combined sequentially to form the equivalent binary number. Hence, the binary equivalent of $3C_{16}$ is 0011 1100_2, which is the same as 11 1100_2 if the leading zeros are omitted.

Hexadecimal digit	**3**	**C**
Binary value	0011	1100

Table 2.17 Converting $3C_{16}$ to binary

[Binary to Hexadecimal]

1. Split binary number into sets of four digits, starting from the right
2. If final set of remaining digits (on the very left) does not have four digits, add leading zeros such that it has exactly four binary digits

3. Use Table 2.16 to map each four-digit group to its corresponding hexadecimal digit
4. Combine hexadecimal digits sequentially to form the equivalent hexadecimal number

[**Hexadecimal to Denary**]

1. Convert each hexadecimal digit to equivalent denary value
    a. Draw table w/3 rows: Place value, Hexadecimal digit, Denary equivalent
2. Multiply each denary value by respective 16^N place values
3. Sum results of each place value

[**Denary to Binary/Hexadecimal**]

1. Division by 2/16
    1.1. Draw table w/3 columns: Denary | Quotient | Remainder
    1.2. Fill in denary in the next empty row
    1.3. Divide the denary number by 2/16 and fill quotient and remainder in same row
    1.4. Repeat steps 1.2 and 1.3, where quotient is the new denary, until quotient is 0
    1.5. Result is remainder column read from the bottom up (i.e. left/bottom → right/top)
2. Sum of Place Values
    2.1. Write out place value of place value (i.e. 2^2,16^3, their denary equivalent etc.)
    2.2. Find highest place value <= denary number
    2.3. Keep subtracting from the highest product place value possible
    2.4. Add digit where necessary
        2.4.1. Binary: 1 under place values that are subtracted, 0 under place values that are not
        2.4.2. Hexadecimal: Number of times highest place value can be subtracted from denary before it becomes less than that highest value

1.2.3 Use two's complement for a fixed number of bits to represent both positive and negative whole numbers in binary

[**Integer overflow**] - Result when an operation produces an integer that exceeds the limits of the representable range

[**Two's complement**] - System used by computers to represent both positive and negative numbers

- First bit always represents number's sign
- Zero is uniquely represented by having all bits set to 0
- Range of numbers that can be represented with N bits is: -2^{N-1} to $2^{N-1} - 1$

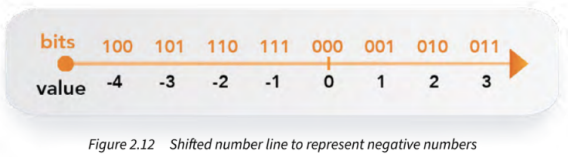


Figure 2.12 Shifted number line to represent negative numbers

Representing negative denary numbers in binary w/ two's complement

1. Flip all bits of the binary form of the positive denary number
2. Add 1

1.2.4 Use the example of an 8-bit extended ASCII encoding for English text to explain how information can be represented as bits for storage or processing by a computer

[**Character (text)**] - A unit of textual data, usually a symbol, letter, digit or space
[**Character encoding**] - The assignment of numbers to characters
[**American Standard Code for Information Interchange (ASCII)**] - A common character encoding method for English characters, including upper-case and lower-case letters

- Extended ASCII standards use 1 byte for each character
    - Represent $2^8 = 256$ distinct characters
    - Can be represented by 2 hex digits, 3 denary digits or 8 binary digits

Using *character encoding*, text is represented on a computer by mapping each character to numbers and storing them as sequences of bits and bytes.

1.3 Logic Gates

1.3.1 Represent logic circuits using either logic circuit diagrams or Boolean statements and convert between the two representations

[**Boolean**] - Data type representing either True or False
0 - False/Off/No
1 - True/On/Yes

[**Logic gate**] - Electrical circuit that performs logical operations based on one or more inputs and produces a single output
[**Logical operation**] - Operation that acts on binary inputs to produce an output according to the laws of Boolean logic
[**Logic circuit**] - Circuit that consists of only one logic gate or multiple connected logic gates
[**Logic circuit diagram**] - Visual representation using symbols to show how the component logic gates are connected together in a logic circuit

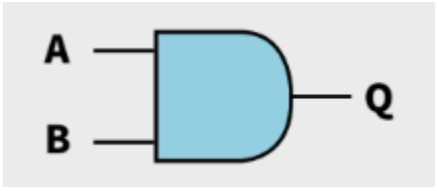
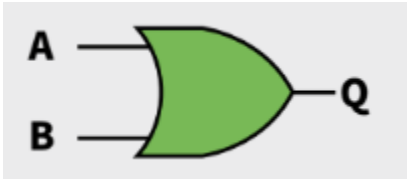
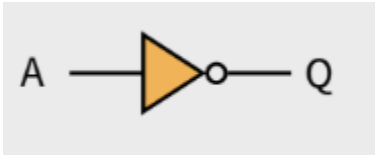
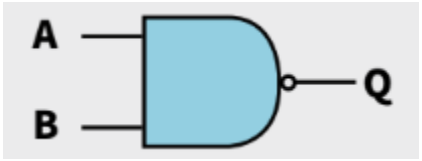
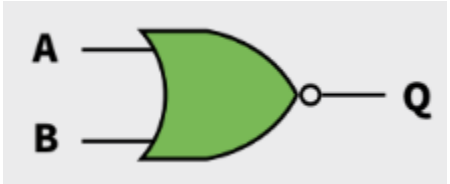
1.3.2 Construct the truth table for a given logic circuit (maximum 3 inputs) and vice versa

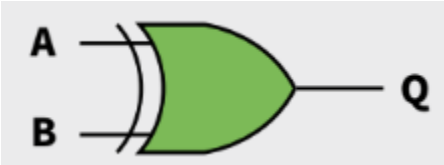
[**Truth table**] - Table showing the resulting output for every possible combination of inputs
>> How to create a truth table:

1. Determine number of rows in truth table: 2^n, where n is number of inputs
2. Fill in input columns in ascending order using binary system (i.e. starting from 000)
3. Output is determined by type of logic gate

converting between truth table, logic circuit diagram and boolean statement can be found in worksheets/class tests i lz type notes for ts

1.3.3 Draw symbols and construct truth tables for AND, OR, NOT, NAND, NOR and XOR logic gates

AND	• $Q = A \cdot B$ 	Both inputs = 1 for output = 1
OR	• OR: $Q = A + B$ 	At least 1 input = 1 for output = 1
NOT	• $Q = \bar{A}$ 	Output is opposite of input
NAND	• $Q = \overline{A \cdot B}$ 	Both inputs = 1 for output = 0
NOR	• $Q = \overline{A + B}$ 	At least 1 input = 1 for output = 0

XOR	• Q = A ⊕ B 	Either input = 1 ONLY for output = 1

1.3.4 Manipulate Boolean statements using the associative and distributive properties of certain logical operators and De Morgan's theorem

[**Order of operations**] (from first → last)
Parentheses → NOT → AND → OR

Property	AND	OR	XOR
Annulment	$A \cdot 0 = 0$	$A + 1 = 1$	
Identity	$A \cdot 1 = A$	$A + 0 = A$	$A \oplus 0 = A$
Idempotent	$A \cdot A = A$	$A + A = A$	
Self-Inverting			$A \oplus A = 0$
Complement	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$	$A \oplus \bar{A} = 1$
Commutative	$A \cdot B = B \cdot A$	$A + B = B + A$	$A \oplus B = B \oplus A$
Associative	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$	$A \oplus (B \oplus C) = (A \oplus B) \oplus C$
Distributive	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $A \cdot (B \oplus C) = (A \cdot B) \oplus (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$	
Absorption	$A \cdot (A + B) = A$	$A + (A \cdot B) = A$	
De Morgan's Theorem	$\overline{A \cdot B} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A} \cdot \bar{B}$	

Table 3.5.1 Common properties of AND, OR and XOR

****De Morgan's Theorem, Associative and Distributive properties are not in given in formula sheet for O-Levels**

1.3.5 Solve system problems using combinations of logic gates (maximum 3 inputs)

Apply logic from 1.3.1-1.3.4 to solve systems ez just practice

Module 2: Algorithms and Programming

2.1 Problem Analysis

2.1.1 For a given problem, identify and remove unnecessary details to specify:

- inputs and the requirements for valid inputs

[**Input (algorithm)**] - Data used by an algorithm that can affect what is required for correct output

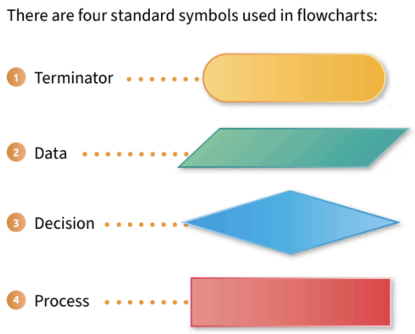
- outputs and the requirements for correct outputs

[**Output**] - Results produced by an algorithm
[**Solution**] - Algorithm that always gives correct outputs when provided with valid inputs

2.2 Constructs

2.2.1 Interpret flowcharts to understand the sequence, selection and iteration constructs

[**Control flow**] - The order in which instructions of a program are run
[**Flow line**] - An arrow that indicates the order in which stages steps should be followed
[**Flowchart**] - A visual representation of an algorithm using symbols to show the flow of a sequence of steps
[**Symbol (flowchart)**] - A special shape to represent an action or step in a flowchart



[**Construct**] - An algorithm structure with exactly one entry point and one exit point
[**Sequence construct**] - A construct for performing multiple instructions in a fixed order
[**Iteration construct**] - A construct for repeated instructions while a particular condition is true

[**Selection construct**] - A construct for choosing between two or more branches based on a particular condition
[**Structured programming language**] - A programming language that encourages or is limited to the use of sequence, selection and iteration constructs that have exactly one entry point and one exit point
[**Branch**] - A sequence of instructions that serves as one option out of two or more choices
[**Iterate**] - To repeat

[**Loop**] - Instructions that are repeated until a condition is met

2.3 Python Code

for the practical part - just have common sense bruh 🙏 practice makes perfect

[**Program**] - Set of instructions written to perform specific tasks on a computer
[**Source code**] - Instructions written in a programming language

[**Command line interface**] - A means of interacting with a program such that commands are given as lines of text
[**Graphical user interface**] - A means of interacting with a program such that commands are given using visual elements such as windows, icons, menus and mouse pointers
[**Syntax highlighting**] - Displaying different parts of source code differently (such as using different colours and fonts) in a text editor based on the rules of the programming language to make reading easier

[**Kernel (Jupyter)**] - A program that runs the code embedded in a notebook
[**Notebook (Jupyter)**] - A file where executable code is embedded in formatted documents that may contain text, images and other media

[**Comment**] - A piece of source code that explains the program but is ignored by Python
[**Literal**] - A value that is represented directly in source code
[**Syntax**] - Rules that determine how the words and symbols in a valid instruction are arranged

[**Identifier**] - A sequence of characters that follows certain rules and can be used as a variable name
[**Initialise**] - To store or assign a value to a variable for the first time
[**Keyword**] - A word that has a special meaning in a programming language and cannot be used as an identifier
[**Statement**] - Code that instructs the computer to perform a specific action

[**Variable**] - A name with an associated value that can be changed while the program is running
[**Expression**] - Code that produces a value

[**Argument**] - An additional value that can be supplied as input to a function call

[**Function**] - A set of instructions assigned to a name that can be used again later
[**Function call**] - The process of assigning arguments to new variables and running the instructions assigned to a function
[**Nested function call**] - Using the return value of one function call as an argument for another function call
[**Return value**] - The output of a function, or the value that a function call is treated as, after the instructions assigned to the function are completed

[**Method**] - A function that is associated with a value
- E.g. value.name_of_method(args)

[**Object**] - A value with associated data and/or methods

[**Binary operator**] - An operator that takes in exactly two values
[**Operand**] - A value that is manipulated by an operator
[**Operator**] - A symbol for performing instructions like a function but using a different syntax
[**Unary operator**] - An operator that takes in exactly one value

[**Floating-point number (**`float`**)**] - A data type to represent real numbers that may contain a fractional part
[**Integer (**`int`**)**] - A data type representing whole numbers
[**String (**`str`**)**] - A data type to represent text as a sequence of characters or symbols
[**List (**`list`**)**] - An ordered sequence of changeable values
[**Dictionary (**`dict`**)**] - A data type to represent an unordered collection of key-value pairs
[**Key**] - A unique value that is used to look up the corresponding value in a dictionary

[**Type casting**] - The process of converting a value from one data type to another

[**Module**] - A collection of variables and functions that need to be loaded before they can be used

[**Escape code**] - A sequence of characters used to input characters that are either difficult to type out as part of a literally or would create a syntax error if not treated specially

[**Replacement field**] - A placeholder denoted by a set of curly braces that will be replaced with a value

[**Immutable**] - Cannot be changed once created
[**Substring**] - A smaller string that appears within a larger string

[**Parameters**] - Special variable in a UDF (user-defined function) that an argument is assigned to when the UDF is called
[**Signature (programming)**] - The first line of a UDF that specifies the UDF's name and its parameters

[**Local variable**] - A variable that is created inside a UDF and can only be used in the UDF's body
[**Global variable**] - A variable that is created outside of a UDF and is readable from the UDF's body of its name is not hidden by a local variable

2.4 Testing and Debugging

2.4.1 to 2.4.5 are **common debugging techniques.
> Using print statements
> Using a trace table
> Backtracking
> Testing incrementally
> Testing small parts of a program

2.4.1 Produce a trace table program by performing a dry run and inspecting the value of variables at each step of a program

[**Trace Table**] - Tabulation of changes in variables and output of an algorithm when performing a dry run
[**Dry Run**] - Process of running through a set of steps manually

2.4.2 Inspect the value of variables at selected steps of a program by inserting print statements.

[**Print statements**] - keep track of how variables change as program runs using print() function

2.4.3 Locate logic errors by backtracking from a point where unexpected behaviour is observed.

[**Backtracking**] - technique of working backwards from location of unintended behaviour to determine its cause (of the logic error)

2.4.4 Test programs incrementally as small additions and changes are made during development.

Test programs as **small additions** and **changes are made during development**

- Tests new features/changes more thoroughly
- Verify small amounts of code added/modified works as intended before moving on

2.4.5 Test small parts of a program by commenting out other parts of the program that are not needed for testing.

Temporarily disabling lines of code to test specifically other pieces of code

2.4.7 Validate input data for acceptance by performing;

Length Check

- Check length of input

Range Check

- Check that input is within certain limits/range of values

Presence Check

- Check that required inputs (needed for program) are provided (i.e. inputted in by user) and not empty

Format Check

- Check that input satisfies additional complex requirements (e.g. following specific format)

Existence Check

- Check that input *exists in a certain collection or repository of data*

Calculation of Check Digit

- A check digit is a numerical value added to a sequence of characters, used to detect errors in data entry or transmission by validating the accuracy of the original data. A check digit is mathematically related to the sequence of characters.

2.4.8 Understand and describe types of program errors; syntax, logic and run-time; and explain why they occur

[**Syntax**] - incorrect source code that violate rules of language, prevents code from running
[**Logic**] - program does not behave as expected, actual output differs from expected output
[**Run-time**] - occurs while program is running, possible result of logic/syntax errors

2.4.9 Design appropriate test cases to cover normal, error and boundary conditions and specify which type(s) of conditions is/are being tested for each test case

Purpose of testing - detect run-time and logic errors
[**Test Case**] - set of inputs and corresponding set of expected outputs for particular program which is used to verify whether program works as intended
[**Testing**] - process by which defects in a program can be detected

[**Normal Conditions**] - input data is expected input during normal use of program
[**Boundary Conditions**] - input is at limit of what program is designed for, or where special handling of input is required

[**Error Conditions**] - input data that should be rejected by program via input validation

2.5 Algorithm Design

[**Algorithm**] - Set of step-by-step instructions for solving a problem or performing a calculation
[**Computational Thinking**] - process of formulating problems so their solutions can be represented as algorithms
[**Decomposition**], 2.5.2 and 2.5.3 - breaking down complex problem into smaller and manageable parts

The following segments are *computational thinking skills*

2.5.2 [Modular Approach] Solve problems by breaking them down into smaller and more manageable parts

- Parts are relatively self-contained
- Solutions for each part is independent of each other
- Different parts are combined to form the final output

2.5.3 [Incremental Approach] Solve problems by solving a small version of the problem and gradually extending the solution to bigger versions of the problem

- Correct outputs are easier to calculate when input is at its lower limits
- Correct output can be calculated when input is incremented

2.5.4 [Generalisation] Use the techniques of solving many small instances of a problem manually to identify the generic steps that are needed to solve the problem in general

[**Generalisation**] - identifying common features from two or more similar items
- Common features can be used to form a single, more general item that can be used to replace or represent the original items

To perform generalisation;
- Common steps that can be used as starting point for generic algorithm
- Repeated steps that can be represented as reusable functions
- Repeated steps that can be simplified/shortened using while or for loops

2.5.5 Recognise when a new problem is similar to an existing problem that has been encountered before and adapt the corresponding solution to solve the new problem

- yuhhh use your brain to see connections between different problems and apply them accordingly

2.6 Software Engineering

[**Software Engineering**] - Process of designing, creating and maintaining software applications

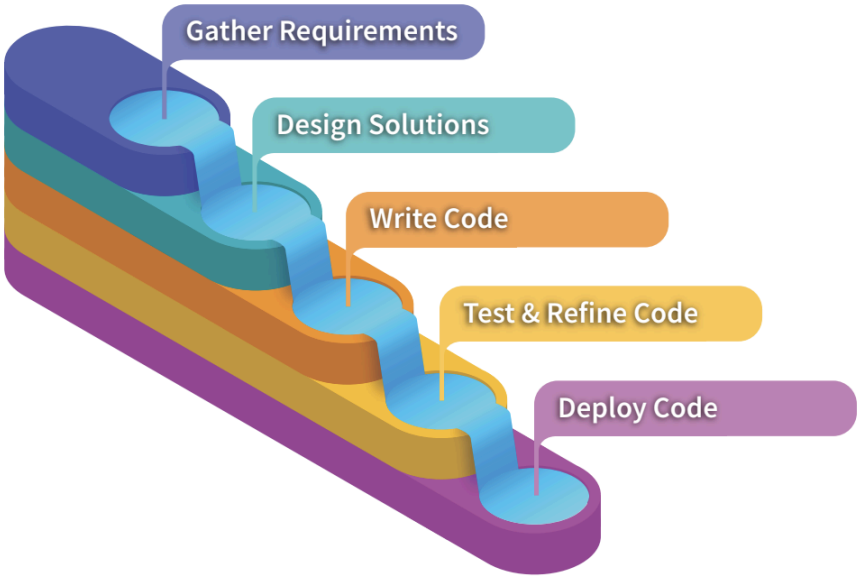


Figure 8.1 Stages in developing a program

2.6.1 Understand and describe the stages in developing a program

	Goal	**Examples**
Gather Requirements	Determine nature of problem, why program is needed and what program is expected to do	>Interview intended audience to understand nature of problem/expectations >Specifying input/output that is necessary for problem
Design Solutions	Consider options available before code is written, choose algorithm based on resources available (e.g. manpower/time)	>Decompose/generalise steps needed to produce required output >Adapting existing solutions

		>Estimating effort needed >Plan possible algorithms w/ flowcharts
Write Code	Write code that performs algorithm as planned in previous stage as efficiently as possible	code lmao
Test & Refine Code	Identify and fix errors or bugs, ensure software functions as intended, improve performance, and enhance reliability and user experience before deployment	>Programmer may make mistakes in translating planned algorithm into code >Solution-planning stage may not have been performed properly >Requirement-gathering stage may have been incomplete
Deploy Code	Actually “roll out” program and used by intended audience	>Training users to use program >Trans. from old program to new program >Evaluating effectiveness of program in solving problem, considering changes that can enhance usability/effectiveness

2.6.2 Recognise that the sequence of software development stages may not be linear and the use of iterative development may sometimes be more appropriate

“Recognise that the sequence of software development stages may not be linear and the use of iterative development may sometimes be more appropriate.” - Mr. Robin Pang 2024

[1] Alternative Methodologies

- Highlights that flexibility is required if issues surfaces in later stages of development

1.1 Agile Development

- Break down large project into smaller increments, allows feedback and changes to requirements and thus can be integrated continuously

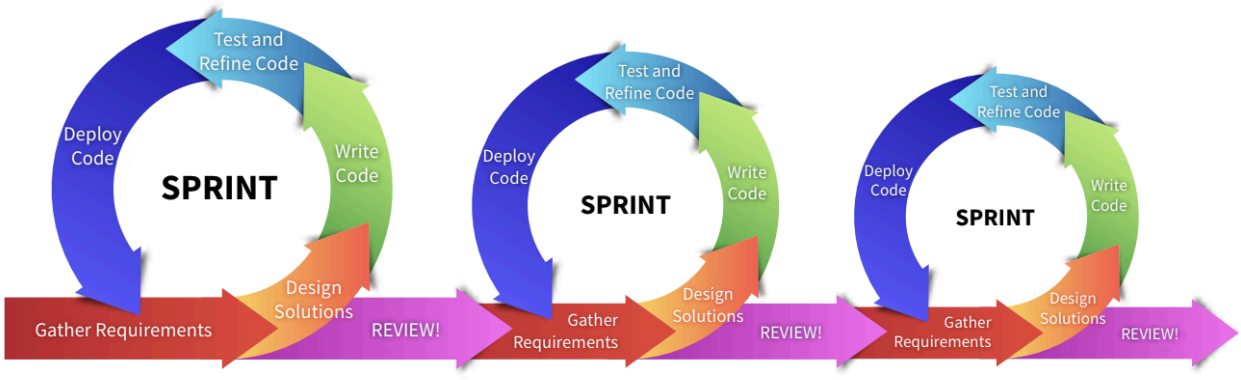


Figure 8.7 Agile is an iterative process where smaller versions of the five stages take place for each sprint

1.2 Test-Driven Development

- Test cases run regularly, code is written incrementally to pass each test
- *Write test cases → Write code → Tidy the code*
- This iterative process tightly integrates these stages for each new feature/functionality that is implemented

Module 3: Spreadsheets

[**Definitions**]:
> **Cell** - The space for data in particular column and row of spreadsheet
> **Spreadsheet** - An electronic worksheet used to manage and manipulate data arranged in columns and rows
> **Cell address** - Location name for a cell formed by its column name and row number
> **Range address** - Location name for rectangular range of cells formed by top-left cell's address, a colon and the bottom right cell's address
> **Formula** - A way to perform calculations on the data in a spreadsheet; may contain functions, operators as well as cell and range references
> **Cell reference** - Description of a cell that can be used in a spreadsheet formula
>> **Absolute cell reference** - A cell reference that does not change when a formula is copied or moved to another cell
>> **Relative cell reference** - A cell reference that changes when a formula containing it is copied or moved to another cell
>> **Mixed cell reference** - A cell reference that has either a fixed column or a fixed row, but not both, using a dollar sign ($) to lock one part
> **Range reference** - Description of multiple cells that can be used in a spreadsheet formula
> **Conditional Formatting** - Spreadsheet feature which automatically formats a cell based on a given criteria

3.1 Program Features

3.1.1 - Appropriate usage of relative, absolute and mixed cell references in formulas
3.1.2 Use *Goal Seek* feature to determine value needed in cell for another cell to reach specified target value

- Function to brute force an answer for you, i.e. adjust a [*given value*] until it reaches [*target value*]

3.1.3 - Use *Conditional Formatting* feature to automatically update cell formatting based on one or more rules

3.2 Functions

Category	Function(s)
Date and Time	DAYS NOW TODAY (DAYS - diff between 2 dates, NOW - date+time, TODAY - date) Note: you can add or subtract days from NOW and TODAY (e.g. NOW()-0.5 returns the date and time from 12 hours before)
Text	CONCAT FIND (case-sensitive, no wildcard characters) LEFT MID RIGHT SEARCH (not case-sensitive, can use any character)
Logical	AND IF NOT OR
Lookup	HLOOKUP (range_lookup - TRUE/omitted=approx. match, FALSE=exact match) VLOOKUP (range_lookup - TRUE/omitted=approx. match, FALSE=exact match) INDEX MATCH (match_type - 1/omitted=largest val<=lookup, 0=exact, -1=smallest val>=lookup)
Mathematical	CEILING.MATH(number, [sig.],[mode]) FLOOR.MATH(number, [sig.],[mode]) MOD(number, divisor) POWER(number, power) QUOTIENT(number, divisor *QUOTIENT is opposite of MOD*) RAND(no arg.s, produces random number between 0 to 1) RANDBETWEEN(bottom, top) ROUND(number, num_digits) SQRT(number), SUM(list of numbers) SUMIF(range, criteria, [sum_range])
Statistical	AVERAGE - returns average from range of values AVERAGEIF - returns average from range of values that satisfy a condition COUNT - counts number of cells within a given range COUNTA - counts number of non-empty cells within a range COUNTBLANK - counts number of empty cells within a range COUNTIF - counts number of cells if the value satisfies a condition LARGE - returns the k^{th} largest value from range of values MAX - returns largest value from range of values MEDIAN - returns median from range of values MIN - returns smallest value from range of values MODE.SNGL - returns mode of a range of values RANK.EQ - returns rank of a value within a range of values, 1 ascending, 0 descending SMALL - returns k^{th} smallest value from range of values

Module 4: Networking

4.1 Concepts

4.1.1 Define computer networks as systems of two or more computers connected by a transmission medium for the exchange of data

[**Computer network**] - A system of two or more computers (or devices) that are connected together by a *transmission medium* for the exchange of data
[**Transmission medium**] - A means of connecting two or more computers together using copper cables, radio waves or light pulses so that they may exchange data and interact with each other

Advantages/Disadvantages of a Computer Network

Advantages	**Disadvantages**
● *Shared storage and resources* ○ A network allows connected devices to make use of shared resources such as printers or files ● *Shared Internet access* ○ Depending on the network's configuration, devices connected to the network may have access to the Internet ● *Shared software* ○ Software can be stored on the central server of a network and deployed to other devices over a network ● *Communication* ○ Devices in the same network are often able to share instant messages and emails for communication	● *Initial costs* ○ Installing a network could be more expensive due to setup and hardware costs ● *Maintenance costs* ○ There are also subsequent costs associated with administering and maintaining the network ● *Security risks* ○ As files are shared through a single network, there is the risk of unauthorised access to data and virus or worm attacks spreading throughout the network, even through a single infected device. ● *Server outages* ○ If the server fails, the network will not be able to function, thus affecting work processes

4.1.2 Describe the difference between wired and wireless transmission media and explain the factors that will determine the use of each medium

Networks can be classified as **wired** or **wireless** based on *transmission medium.*

[**Ethernet**] - Most commonly used wired network standard for local area networks
[**Wired network**] - A network of devices connected by physical cables
[**Wi-Fi**] - Most commonly used wireless network standard for local area networks

[**Wireless network**] - A network of devices in which signals are transmitted without the use of physical cables

Wired vs. Wireless

Factor	Wired	Wireless
Cost	• Initially cheaper but becomes more expensive as the network grows due to the cost of cables	• Initially expensive due to the cost of wireless networking hardware but becomes more cost-effective as the network grows
Speed of transmission and bandwidth	• Faster and higher bandwidth as cables provide dedicated connection	• Generally slower and lower bandwidth due to possible interference from radio waves or microwaves; varies according to user location in relation to the network
Reliability	• More reliable as data transmission is less affected by interference	• Less reliable due to potential interference from radio waves and microwaves or blockage from physical obstructions
Security	• More secure as the network is less susceptible to interception and hacking from a distance	• Less secure due to possible intrusions by hackers who do not need physical access to a cable or network hardware
Mobility of users	• Lower as network connections are fixed at specific spots and users cannot move to other locations	• Higher as users can move about freely within the range of the wireless network
Scalability	• More cumbersome to add new devices to the network due to the need for additional cables	• Easier to add new devices to the network as no additional hardware is usually needed; however, there is a limit as interference increases with the number of wireless devices
Physical organisation	• Tends to look more disorganised due to cables running across floors	• More organised without cables

4.1.3 Differentiate between Local Area Networks (LANs) and Wide Area Networks (WANs) based on their geographic scope

>> *geographical coverage* of network is main factor in classifying into LAN or WAN
[**Local area network**] - A network of computing devices connected within a small geographical area, typically within the same building, such as a home, school or office
[**Wide area network**] - A network of computing devices covering a large-scale geographic area, typically across multiple geographical locations

- Consists of multiple smaller networks such as LANs
- Not limited to single geographical location unlike LAN, can span long distances via long-range transmission media such as telephone lines, fibre optic cables or satellite links
- Can be private or public. Private is built + maintained by large multinational companies and government organisations. The Internet is a public WAN.
- More expensive and higher-speed technology than a LAN

4.1.4 Compare and contrast the client-server and peer-to-peer network architectures in terms of purpose, organisation and bandwidth

[**Client**] - A device that initiates a connection to a server to request for resources and services
[**Server**] - A device that shares resources with and responds to requests from *clients*

- Higher capacity and is more powerful than normal client devices as it needs to manage resources and services

[**Bandwidth**] - The average number of bits of data that can be transmitted from a source to a destination over the network in one second; usually measured in megabits per second (Mbps) or gigabits per second (Gbps)

Feature	**Client-Server**		**Peer-To-Peer**	
Pros/Cons	+ve ● Centralised control of data and resources ● Easy to schedule backups of all shared files at regular intervals ● Security may be enhanced with the use of specialised software or operating system features that are designed for servers	-ve ● Higher initial cost due to the need for a server ● Administrative costs needed for the maintenance of server and clients	+ve ● Cheaper to set up as there is no cost related to dedicated servers ● Easy to set up as no specialised software or operating system features are needed	-ve ● More effort is required to access and back up resources as they are stored locally within each computer instead of centrally in a server ● Security is an issue as access rights are not administered by a central server
Function	● Data and resources are shared using one or more dedicated servers; each device has a distinct role - client or server		● Data and resources are shared directly between devices; each device acts both a client and server	
Organisation of hardware	● Each client is connected to one or more dedicated servers		● Each device in the network can serve as a client and server at the same time	

Bandwidth	• Typically high as servers have centralised control to use *bandwidth* more efficiently and offer more reliable connections, but this also depends on the capabilities of the servers	• Varies depending on how data needs to be transmitted; bandwidth may be reduced if a single device must handle many large requests or if requests must rely on a device with an unreliable connection, but bandwidth may also be increased if large requests can be divided into smaller requests that are handled by multiple devices simultaneously
Security	• High as access rights can be controlled centrally at a server	• Low as security is handled by each device and not by a central server
Setup cost	• High as the use of specialised high-performance servers would be needed	• Low as basic devices can act as servers to share resources
Storage	• Centralised and usually managed at the server by a network administrator	• Decentralised and usually managed by individual users at each device
Application	• Found in businesses or organisations with many users	• Found in homes or small businesses where there are fewer users

4.1.5 Identify and state common applications of star and mesh topologies in a home network

[**Topology**] - The physical layout of a network in terms of devices and connections
[**Star Topology**] - Network hardware such as a switch or wireless access point is centre of network with connections to all other computers, with computers sending data to central network hardware and hardware forwards data to intended destination
[**Mesh Topology**] - Devices are connected to each other and cooperate to relay messages to their intended destination

- Fully-connected mesh
    - Every device directly connected to every other device
- Partially-connected mesh
    - Some devices connected to several others, but not to all

Factor	**Star Topology**	**Mesh Topology**
Applications in Home Networks	• Suitable for both wired and wireless home networks	• Typically only suitable for wireless home networks, especially home automation
Complexity	• Less complex	• More complex
Cost	• Typically lower cost depending on medium; only one piece of central network hardware is needed but cables	• Typically higher cost; more network hardware needed

	for a wired network can be costly as the network grows	
Robustness	• Less robust as the central hardware is a single point of failure that will stop the entire network if it fails	• More robust as there would be typically be multiple different routes between devices, and the network would continue to function if any one device or hardware fails
Range	• Typically smaller; limited by cable length and distance to the central hardware	• Typically larger; each addition of network hardware extends the network's range without sacrificing speed
Standards	• Highly standardised so hardware from different companies can usually work together	• Currently not highly standardised so hardware from different companies may not work together

[**Advantages vs. Disadvantages**]

Topoly	Advantages	Disadvantages
Mesh	High reliability and resiliency, redundant paths, decentralized, self-healing, high scalability	More complex to set up and manage, higher cost, more nodes may lead to more congestion
Star	Simple to set up and manage, centralized, easy to add new nodes	Single point of failure, less reliable, less scalable

4.1.6 Define protocols as standards and rules that govern communication over a network

[**Network protocol**] - A set of standards and rules that govern how two or more devices communicate over a network
[**Packet (computing)**] - A unit of broken-up data containing a header with information that are needed for transmission, such as the source and destination addresses
[**Packet number (or sequence number)**] - A number in a packet's header that indicates the position of that packet in the sequence of data being transmitted

4.1.7 Explain that LANs typically use protocols where data is transmitted as individual packets

For protocols used to communicate over the Internet,
> Data sent is broken up into smaller *packets* before being sent independently over network through hardware such as switches and routers
> Each packet containers header w/ information about source needed for transmission
> Each header contains *packet number*, indicating position packet in sequence of data being transmitted

> Data packets are reassembled in packet number order at destination device

4.1.8 Explain the use of parity, checksums, echo checks and automatic repeat requests for detecting errors in packet transmission

Data can be lost or corrupted during transmission.

Parity Bits

[**Parity bit**] - An additional bit, which can be either 0 or 1 depending on whether an odd or even parity system is used, that is appended or prepended to a string binary data before transmission
[**Parity check**] - An error-checking technique which uses a parity bit to detect errors

Odd parity system - Total number of 1-bits in each packet (inc. parity bit) is odd
Even parity system - Total number of 1-bits in each packet (inc. parity bit) is even

Checksums

[**Checksum**] - A calculated value that is used to determine the integrity of transmitted data

A *checksum* is mathematically related to data such that modifications to data would break said mathematical relationship and hence be detected (e.g. NRIC check thingymajiggy).

Checksum is calculated before transmission and sent with data. Checksum is recalculated and compared with the original checksum value, if they differ, an error has occurred. The recipient can then request for data to be sent again.

Checksum	Check Digit
Similarity: Both are mathematically related to the data such that modifications to the data would break this mathematical relationship and hence be detected.	
Differences	
More complex	Less complex
Ensures the integrity of **larger** pieces of data during **transmission**	Guards against errors when **manually** entering **short** sequences of digits

Echo checks

[**Echo check**] - An error detection method where the receiver sends a copy of what was received back to the sender

It's basically checking with your friend that what they said was actually what they said.

Advantages - Detects more kinds of errors compared to parity bits and check sums
Disadvantages

- Checks roughly double the amount of traffic and time needed to complete each transmission

- W/o further checks, sender cannot tell if error occurred during initial transmission or when echo was transmitted
- By itself, echo check does not address a situation where data is lost and the receiver receives no data at all

Automatic repeat requests

[**Automatic Repeat Requests (ARQ)**] - An error detection and correction method where the sender automatically resends the data if no acknowledgement from the receiver is received within the timeout period
[**Timeout (ARQ)**] - A time limit used by the sender to wait for an acknowledgement before it is assumed that the data or acknowledgement itself was lost in transit

Other checks are for data corruption only, ARQ addresses the issue of packets being lost in transmission and data corruption as well.

1. **ACK Received (**sent from receiver**)**: Indicates data was received without errors
2. **NACK Received (**sent from receiver**)**: Indicates data was corrupted and needs to be resent
3. **No Acknowledgement Received (Timeout Occurs) (**nothing is sent from receiver**)**: If neither ACK nor NACK is received within timeout period, sender assumes data was lost or acknowledgement itself was lost in transit. Data is resent.

Advantages - Addresses and recovers situations where data is lost and receiver never receives data at all
Disadvantages

- Only deals with packet loss; needs to be used together with negative acknowledgements and some form of error detection such as parity bits, checksums or echo checks to detect data corruption
- Resending data when a timeout occurs wastes bandwidth and increases complexity if the data was actually received without errors by the receiver and the positive acknowledgement was just lost or delayed; the receiver must be able to identify and ignore duplicate packets from the sender to handle this possibility

4.2 Home Networks and the Internet

4.2.1 Explain that home networks are examples of LANs and that the internet is an example of a WAN that is formed by connecting many different LANs from around the world together.

Internet (*WAN*) → Home networks (*LAN*) → Computers
→ Company networks (*LAN*) → Server for a website

Data is transmitted between LANs using *routing devices*.

To connect to these routing devices and access the Internet, a home user typically needs to subscribe to a service provided by an *Internet Service Provider (ISP)*. The ISP facilitates this connection by providing the *hardware* needed for a *gateway* to communicate with devices that are not in your home network.

[**Gateway**] - A device that serves as an entry and exit point for communication with devices that are not in the local network
[**Internet Service Provider (ISP)**] - An organisation that offers services for accessing the Internet

[**Common network hardware**]

- Modems
- Network Interface Controllers
- Network Switches
- Wireless Access Points
- Routers

4.2.2 Explain that modems are used to provide internet access by converting from the protocols used by an Internet Service Provider (ISP) to the protocols used by LANs.

[**Modulation**] - Conversion of digital data into a form suitable for transmission
[**Demodulation**] - Conversion of transmitted signals into digital data
[**Modem** (haha get it, MODulation and DEModulation)] - Device responsible for *modulation* and *demodulation*

Modems are used over Ethernet cables and Wi-Fi when it comes to connecting data over a long distance. ISPs use specialised protocols and long-range transmission media (telephone lines, cellular radio waves or fire optic cables) and modems are used as the translator to send data over the Internet and receive data from the Internet.

4.2.3 Explain that computers use network interface controllers to communicate via different transmission media.

[**Network Interface Controller (NIC)**] - Provides the hardware interface to enable transfer of data between a device and a network

- Each NIC has a unique 480bit MAC address

Any device connecting to a network needs an NIC, and can use Ethernet cables or wireless means using Wi-Fi.

4.2.4 Explain that Media Access Control (MAC) addresses are used by network interface controllers, network switches and wireless access points to direct data within the same LAN while Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) addresses are used by routers to direct data across different LANs.

[**Network switch**] - Constructs a single network by connecting multiple devices together

- In home networks, they are used to construct a LAN using Ethernet cables

[**Wireless access point (WAP)**] - Network hardware that lets wireless devices connect to a wired network

- Typically has Ethernet connection to either switch or router and allows wireless devices to join network using Wi-Fi

[**Service set identifier (SSID)**] - A 32-byte string that identifies a Wi-Fi network

[**Router**] - Network hardware that forward packets between separate networks

A switch connects the devices together and forms a network, the router is the bridge between LANs, i.e. it connects separate networks together.

4.2.5 Compare and contrast MAC, IPv4 and IPv6 addresses in terms of purpose, bit length, degree of permanence and typical representation formats.

[**Network address**] - A unique name in the form of a sequence of bytes

- For computers to locate each other and transmit data to the correct destination
- Similar to actual mail, you need to know the other person's address to send them information

IP - Internet Protocol

Feature	**Media Access Control (MAC)**	**IPv4**	**IPv6**

Purpose	• Transmission between devices in the same LAN
	  ○ Basically unique address for each device
		• Reflect network's topology such that devices in the same network location have similar IP addresses
		  ○ Hierarchical
		  ○ Basically address for each network
			• Maximum number of possible IPv4 addresses is 2^{32}, i.e. 4.3 billion. Rapid growth of Internet meant this number would not be sufficient
			• Developed to solve the issue of IPv4

Bit length	• 6 bytes, i.e. 48 bits	• 4 bytes, i.e. 32 bits	• 16 bytes, i.e. 128 bits
Degree of permanence	• Permanent	• May change each time a device connects Internet to match network's routing prefix	

Typical representation formats	• NN:NN:NN:DD:DD:DD
	• Sequence of 6 hexadecimal numbers, one for each byte of address separated by hyphens or colons
	• NN:NN:NN
	  ○ Manufacturer's identity number
	• DD:DD:DD
	  ○ Device's serial number
		• Sequence of 4 denary numbers, one for each byte of address separated by dots
		• Value of 1 byte can vary from 0 to 255, thus each denary number must be between 0 to 255 inclusive
			• Sequence of 8 hexadecimal numbers, one for each 4 bytes of address separated by colons
			• Each group consists of a 4-digit hexadecimal ranging from 0000_{16} to $FFFF_{16}$
			• For compactness, leading zeros can be omitted

These methods of shortening IPv6 addresses are illustrated in Figure 10.28.

Figure 10.28 Shortening an IPv6 address

However, no more than one such section of consecutive zeros can be replaced with double colons. For instance, 2001:db8:0:0:1:0:0:1 can be written as 2001:db8::1:0:0:1 or 2001:db8:0:0:1::1, but not 2001:db8::1::1. Figure 10.29 illustrates this.

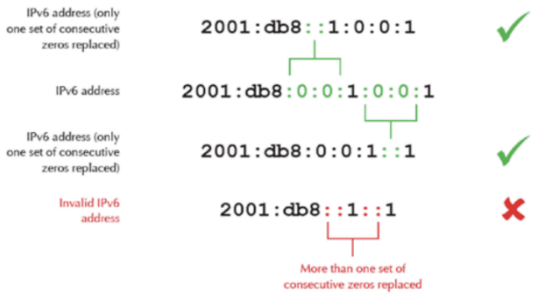


Figure 10.29 Only one section of consecutive zeros can be replaced in an IPv6 address

•

4.2.6 Connect a router, a switch and a wireless access point to a modem correctly such that multiple computers form a LAN that can access the internet via wired and wireless transmission media.

Figure 10.31 shows how a modem, a router, a switch and a wireless access point may be connected to set up a home network, to intergrate both wired and wireless devices. In this example, the modem is connected to a fibre optic outlet that provides Internet access from the ISP via optical fibre. The modem is then connected to the Intranet/WAN socket of the router.

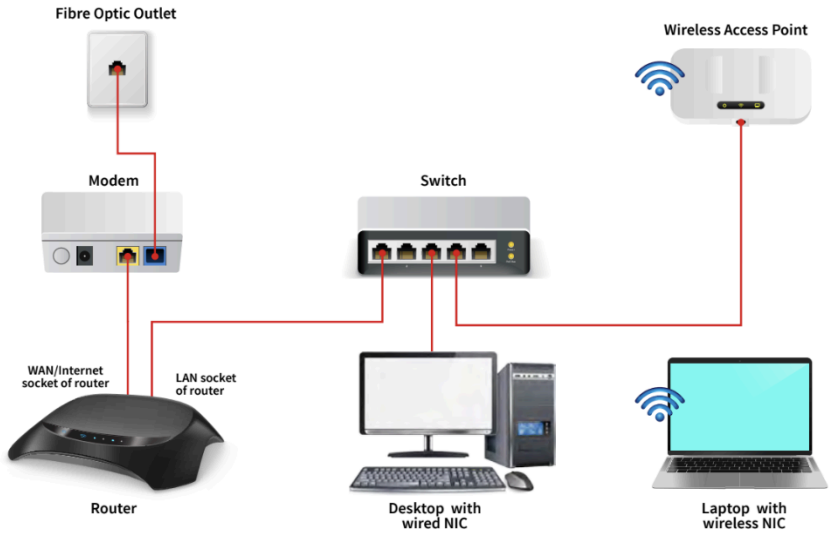


Figure 10.31 Setting up a modem, a router, a switch and a wireless access point to form a home network

Instead of having separate devices, it is also possible to use a "wireless router" that effectively functions as a router with a built-in wireless access point. The multiple LAN ports on the router also function as a built-in switch. This allows the home network to be simplified further, as shown in Figure 10.32.

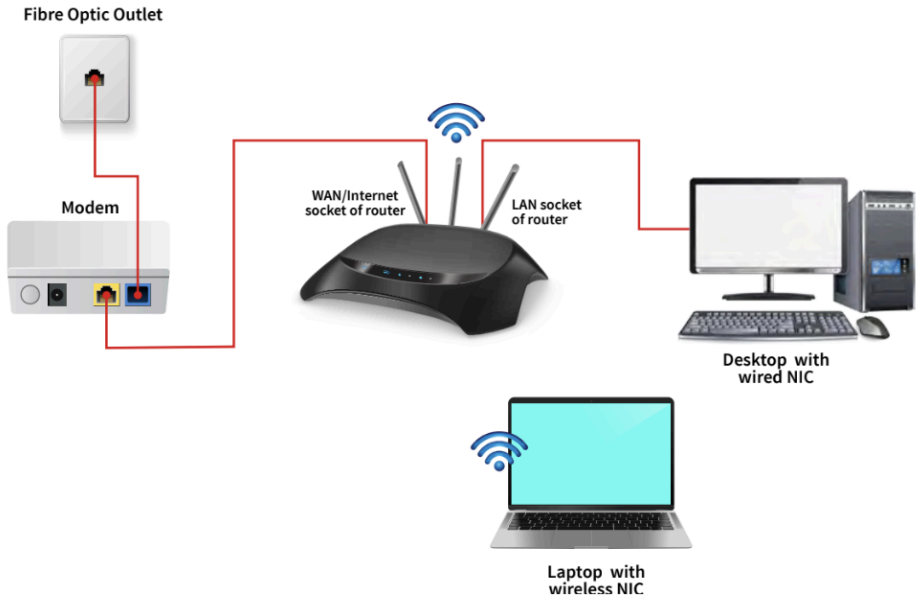


Figure 10.32 Simplified home network using a wireless router

4.3 Security and Privacy

4.3.1 Compare and contrast security and privacy in terms of what kind of data is being protected, what the data is being protected from and how that protection is enforced.

[**Confidentiality**] - Protection from unauthorised access
- Data owned by someone is used by someone else

[**Integrity**] - Protection from unauthorised modification
- Data is intentionally or accidentally altered or made incomplete without permission

[**Availability**] - Ability to access the data in a timely and uninterrupted fashion
- Data may become inaccessible to authorised users
- [**Data corruption**] - Introduction of errors and inconsistencies in data
- [**Denial-of-service attack (DOS)**] - A type of attack where a server is overwhelmed with excessive requests such that it cannot respond or responds very slowly to legitimate requests

[**Personal Data**] - Any data related to a person that may allow that person to be identified

Factor	**Security**	**Privacy**
Definition	• Practice of protecting the confidentiality, integrity and availability of data	• Practice of protecting the confidentiality and control of data
Kind of Data Being Protected	• Data in general	• Personal data specifically
What Data is Protected From	• Unauthorised access + modification, reduced ability to access data • Malicious and/or illegal acts	• Unauthorised access, reduced ability to control how personal data is used or shared • Acts that user may not consent to even if they are not malicious or illegal
How Protection is Enforced	• Technical protections • Legal protections (e.g. Computer Misuse Act, Personal Digital Protection Act) • Careful adherence to best practices and configuration of security settings • Careful understanding of privacy policies and configuration of privacy settings	

4.3.2 Explain how human actions threaten security and privacy by causing data corruption (through physical or non-physical means) or exposure of private data.

Means	Threat of Human Actions to Data Integrity
Physical	*Data corruption due to:* ● Exposure of storage media to physical impact or extreme temperatures ● Not ejecting storage device properly before physical removal
Non-physical	● Overwriting of data due to multiple users working on the same file that is shared on a network ● Data corruption due to incorrect data entry

4.3.3 Explain how anti-malware programs enforce security and privacy by preventing malware from running and removing malware that may be present on a computer.

[**Anti-malware**] - Software to detect, remove and stop malware from running
[**Signature**] - Identifying characteristics that are used to detect a known version of some malicious software

- List of identifying characteristics needs to be updated regularly to ensure protection provided by programs continues to be effective against new malware
- Usually updated automatically through Internet for convenience

4.3.4 Explain how firewalls enforce security and privacy by using either hardware or software to monitor packets and decide which packets should be permitted or blocked based on a set of configurable rules.

[**Firewall**] - Device or computer program that prevents unauthorised access to or from a network

- Confidentiality/Integrity
- Monitors each piece of data transmitted through a network
- Data is either blocked or pass through, based on set of rules configured by an administrator

4.3.5 Explain how encryption enforces security and privacy by making encrypted data appear meaningless without the corresponding secret key.

[**Encryption**] - Process of encoding a message so that a secret key is needed to decode the data

- Confidentiality
- Prevents unauthorised access by only allowing authorised users to have the secret key

4.3.6 Explain how the Personal Data Protection Act (PDPA) enforces privacy by legally requiring that organisations do the following when collecting personal data:

[**Personal Data Protection Act (PDPA)**] - A law that governs the collection, protection and use of personal data in Singapore

- The points below fulfil a user's confidentiality
- seek consent from the individual
- disclose the purpose for collecting data when seeking consent and
- retain the data for only as long as necessary to fulfil the stated purpose.

By complying with the points above, third-party services/users are unable to indirectly violate the privacy of victims. To ensure compliance, organisations found to be in **violation of the PDPA may be required to pay heavy fines**.

4.3.7 Explain how adware threatens security and privacy by installing itself without the user's knowledge and displaying unwanted advertisements

[**Adware**] - Type of malware that installs itself without the user's knowledge and displays unwanted advertisements

- Integrity

[**Malicious software**] - Software that is intentionally used to damage, disrupt or gain unauthorised access to a computer system

4.3.8 Explain how spyware threatens security and privacy by secretly collecting personal information and transmitting this information to attackers without the user's knowledge.

[**Spyware**] - Type of hidden malware that secretly collects personal information about its users and transmits this information to attackers without the users' knowledge

- Confidentiality

4.3.9 Explain how cookies are typically not malicious but can threaten privacy by tracking a user's browsing history across multiple web sites.

[**Cookies**] - Small pieces of data stored by the web browser when a user visits a website

- Needed to keep track of authentication information to identify currently logged in users
- BUT can *keep track of user movements and preferences within website* and advertising companies can use cookies to *keep track of users as they move from one website to another*. Thus cookies can be *a threat to data privacy*
- Confidentiality

4.3.10 Explain how phishing threatens security and privacy by using emails and fake websites that appear to be from reputable companies to steal personal information.

[**Phishing**] - Use of emails, messages and fake websites that appear to be from reputable companies to steal personal data from users

- Confidentiality

4.3.11 Explain how pharming threatens security and privacy by intercepting requests to legitimate websites and redirecting them to fake websites while still appearing to use the same address as the legitimate website.

[**Pharming**] - Interception of requests sent from a computer to a legitimate website and the redirection of those requests to a fake website to steal personal data

- Confidentiality

4.3.12 Describe good computing practices that can mitigate the threats posed by adware, spyware, cookies, phishing, pharming and human actions.

1. Use Strong Passwords
    a. [**Authentication**] - The process of verifying the identity of a user
2. Use Multi-Factor Authentication
    a. [**Authentication factor**] - A category of evidence that is used for authentication:
        i. Something users knows
            1. [**Security token**] given by banks/organisations to users accessing accounts online
                a. Device used specifically for authentication purposes
        ii. Something user owns
            1. [**Two-factor authentication**] like an OTP generated from security token or mobile phone that user owns
                a. Type of authentication that uses evidence from both something the user knows and something the user owns
        iii. Something unique measured from physical part of user
            1. [**Biometrics**] - Type of authentication based on measurement of human physical characteristics
                a. Thumbprint
                b. Face
                c. Voice
3. Identify Scam Attempts
    a. Requests for personal data
        i. cuz what reputable company would do that lol
    b. Errors (inaccurate info/grammar)
    c. Generic greetings

    d. Sus email address
    e. Sus links
    f. Urgent requests
4. Update Software Regularly
5. Manage Cookies
6. Make Regular [**Backups**] - Copy of data that is made so it can be recovered if the original is lost

4.3.13 Analyse the effects of anti-malware programs, firewalls, encryption and the PDPA against the threats posed by adware, spyware, cookies, phishing, pharming and human actions.

[**Summary of effectiveness of the following programs against malware**]

1. Highly Effective
2. Somewhat Effective
3. Not Effective

<table>
<tr><th></th><th>Anti-Malware</th><th>Firewalls</th><th>Encryption</th><th>PDPA</th></tr>
<tr><td>Adware</td><td rowspan="2">Can detect and remove adware + spyware</td><td>Can prevent some adware from downloading advertisements</td><td>Does not typically prevent adware from functioning</td><td rowspan="2">Adware and spyware are illegal and likely to ignore legal protections such as the PDPA</td></tr>
<tr><td>Spyware</td><td>Can prevent some spyware from sending out collected data</td><td>Can prevent some collected data from being exploited</td></tr>
<tr><td>Cookies</td><td>Does not typically block or clear cookies</td><td>Some can block cookies</td><td>Does not typically prevent cookies from tracking users</td><td>Restricts how cookies can be used to collect personal data</td></tr>
<tr><td>Phishing</td><td colspan="2">May block access to a blacklist of known phishing sites</td><td rowspan="2">Does not help if users give data willingly to a phishing/spoofed site</td><td rowspan="2">Phishing/pharming is malicious and likely to ignore legal protections such as the PDPA</td></tr>
<tr><td>Pharming</td><td>Can stop some forms of pharming that use malware</td><td>May block access to a blacklist of known “fake” IP addresses</td></tr>
<tr><td>Human Actions</td><td>Can prevent installation of malware as a result of carelessness</td><td>Can prevent malicious network intrusions by attackers</td><td>Prevents data from being exploited if it is accidentally exposed</td><td>Restricts the collection of personal data and punishes violations</td></tr>
</table>

Module 5: Impact of Computing

5.1 General

5.1.1 Give examples of the impact of computers in the following industries:

- Communication: ability to connect people and businesses over long distances
    - Digitalisation
        - Digital transmission > analogue transmission (analogue noise lose clarity, digital noise can still be decoded clearly)
    - Real-time multimedia communication
        - Limited speed and types of media → Long distances with combination of various media (digital transmission allows more efficient compression algorithms and communication protocols)
    - Mobile computing
        - Portability of computing and communication devices (e.g. laptops, smartphones)
    - Accessibility and globalisation
        - Internet enables interaction between diverse cultures and sharing of ideas, allowing for seamless exchange of information across borders
        - AI automates transcription and translation of messages into different languages with remarkable speed and accuracy
    - Social media and misinformation
        - Allows anyone with Internet access to produce content
        - Used to reinforce existing opinions or spread rumours and misinformation
    - Security and privacy risks
        - Easier to break confidentiality of messages due to digitalisation of communication
        - Easier to track online movements and preferences of users, may be exploited at the expense of privacy of individuals
- Education: easy access to online classes and large amounts of information via the internet
    - Communication and collaboration
        - New tools from computing used to communicate and collaborate for learning without needing to be in same location or online simultaneously
    - Access to information
        - Books in a library → Vast amounts of information from the Internet
    - Personalised learning
        - Static learning materials → Digital learning materials, lessons being more readily adjusted and customised for individual students' needs
    - Interactive and multimedia tools

        - Static text and images → Videos, interactive simulations and virtual environments: More engaging and relevant lessons to students
    - Distance and online learning
        - Lessons conducted via mail (significant delays from receiving, submitting and receiving feedback for assignments) → Carried out via digital communication and collaboration tools: Enables students to attain educational qualifications regardless of geographical constraints
    - Assessment and feedback
        - Manually grading takes much effort to customise tasks and provide individual feedback → Certain assignments can be automatically graded, providing immediate feedback
        - AI can adapt to student's skill level when curating questions
    - Need for digital literacy
        - Revolutionised society and j*b market, necessitating countries to revise their education systems
        - Schools now must teach digital literacy skills essential for everyday life and the contemporary workplace
        - Some teachers are upskilling themselves and assume responsibility of developing digital literacy of students
- Transportation: widespread access to navigational services via Global Positioning System (GPS) and emergence of self-driving vehicles
    - Navigation and GPS
        - Printed maps (inconvenient and static) → Detailed digital maps and real-time information on travelling conditions are readily accessible through mobile computing
        - GPS utilises satellite data to determine user's location
            - Easily obtains navigational directions to get from one location to another in real-time
            - Requires Internet and electricity
    - Smart and autonomous vehicles
        - Offers convenience and may safer than human drivers
        - Ethical concerns over moral choices (should AI prioritise safety of drivers or pedestrians? who is responsible for accidents, vehicle's owner or software developer?)
    - Traffic monitoring
        - Manual methods → GPS enabled mobile devices automate this process, offering more accurate traffic condition estimates in real-time
        - Privacy concerns regarding misuse of location data
    - Enhanced public transportation
        - Contactless payments replacing physical tickets, automatic calculation of fares, real-time information on schedules and delays are readily available
        - Simplified process of hailing taxi

- Retail: more reliable tracking of available stock and emergence of self-checkout counters
    - Inventory management
        - Barcode labels to uniquely identify each product with specialised cameras enables stocktaking to be performed more quickly and and reliably with the aid of computers
        - Near-Field Communication tags are electronic devices are similar to barcodes, but more efficient as they can be read wirelessly using a reader device: inventory management can be streamlined further, allowing for real-time tracking of stock levels and reducing likelihood of stocktaking errors
    - Checkout process
        - Total cost of purchased items can be quickly determined without manual entry of prices
        - Cashless payments offer greater convenience but increased risk of fraud
        - Automated checkout kiosks reduce wait times and need for cashier staff but inconveniences customers and increased risk of shoplifting
    - E-commerce
        - No longer limited to making purchases solely in physical stores
        - Accessible outside of regular business hours, offering wider range of products
        - Track purchase histories for personalised recommendations
        - Physical stores may be preferred to physically inspect items or save on shipping costs
        - Security and privacy concerns regarding information collected by e-commerce sites

5.2 Intellectual Property

5.2.1 Define intellectual property as creations of the mind that have value but can exist purely as data.

[**Intellectual Property**] - Creations of the mind that have value but can exist purely as data with no physical form

- [**Originality and Creativity**] - Work must be original and showcase a degree of creativity and cannot be copied from existing works
- [**Fixation**] - Work must be fixed in a tangible medium of expression and cannot be an idea without a fixed and recorded description
- [**Ownership**] - Work must have an identifiable owner or author for legal rights to be assigned and enforced
    - In Singapore, humans can be recognised as authors under the Copyright Act

5.2.2 Describe copyright as the legal right of owners to control the use and distribution of their intellectual property under the Copyright Act.

[**Copyright**] - Legal right of owners to control the use and distribution of their intellectual property
[**License**] - Official description of activities that are authorised or forbidden by by the owner of intellectual property

5.2.3 Explain that copyright owners can grant a license to authorise or forbid the use and distribution of their intellectual property under certain conditions.

Licenses can be written for scratch, or re-used from a wide range of existing licenses with different levels of restrictiveness.

5.2.4 Identify computer programs as examples of intellectual property and distinguish between proprietary software, freeware, shareware as well as free and open-source software (FOSS) based on their licenses.

[**Free and open-source software (FOSS)**] - Software where users are given freedom to change, copy, study and share the software and its source code
[**Proprietary Software**] - Commercial software for which most of the legal protections under copyright are retained

- [**Shareware**] - Demonstration software that is distributed for free but for a specific evaluation period only
- [**Freeware**] - Proprietary software that is available for use at no cost

5.2.5 Recognise software piracy as the illegal use and distribution of copyrighted computer programs in a manner that is forbidden by their license.

[**Crack**] - Program that modifies software so that the software cannot detect that it is being used illegally
[**Software Piracy**] - Crime copying, distributing and/or using copyrighted software in a manner that is not permitted by its license

- Example of [**Copyright Infringement**] - Use or distribution of copyrighted work without permission of copyright owner
    - When using materials from Internet for work, try to adhere to the following guidelines:
        - Check and follow website's terms and conditions
        - Limit reproduction of a copyrighted work to 10% (for educational purposes only)
        - Consider using public domain material instead
        - Verify and cite your sources when using generative AI

5.3 Communication

5.3.1 Explain why the promotion of social media posts based on engagement rate helps to deliver relevant content to users but can also lead to the proliferation of falsehoods.

[**Engagement rate**] - Measure of how much other users actively interact with a social media post

- Social media sites promotes post based on engagement rates for:
    a. Increase site's relevance to users
        - Prioritising posts with high engagement ensures users seeing posts that are more likely to be of interest, helping social media sites deliver more relevant content, increasing sites' potential value and usefulness to users
    b. Incentivise creation of relevant posts
        - Prioritising posts with high engagement rewards creators that making interesting content with more visibility and larger audience, incentivising them to produce content users are interested in and likely to engage with
    c. Increase advertising revenue
        - Main source of revenue for most social media sites is advertising
        - Engagement is evidence users are paying attention to a post and social media sites can charge advertisers higher rates for placing ads around content with high engagement rates

[**Feed (social media)**] - Feature of social media sites where posts from other users are displayed as a continuously updated stream of content

[**Post (social media)**] - Any content (e.g. text, pictures, videos, links) shared by a user on social media

5.3.2 Explain how the Protection from Online Falsehoods and Manipulation Act (POFMA) enables the government to tackle the spread of fake news by:

- establishing fines and/or prison terms for engaging in prohibited activities
- optionally requiring offenders to put up a correction notice or to take down the falsehood and
    - Issued as a "Correction Direction" in most cases
        - Put put correction notice against original post and provide warning about falsehood and clarification of facts
    - More serious cases issue offenders with "Stop Communication Direction"
        - Take down original post with falsehood
- identifying sites that repeatedly spread falsehoods.
    - Act allows government to blacklist such sites
    - Identified as "Declared Online Locations"

POFMA was passed by the Parliament of Singapore in 2019 to tackle the spread of fake news. This law specifically targets false statements of fact that are communicated in Singapore through the Internet and are against the public interest. It is not intended to cover opinions, criticisms, satire or parody.

5.4 Emerging Technologies

5.4.1 Describe Artificial Intelligence (AI) as the ability of a computer to perform complex tasks without constant human guidance and improve its performance as more data is collected.

[**Artificial intelligence (AI)**] - Ability of a computer to perform complex tasks without constant human guidance and improve its performance as more data is collected

5.4.2 Given examples of common personal and business tasks that can be performed well by AI: face recognition, voice recognition, image classification and spam filtering.

- Face recognition
    - Security system authentication
    - Media applications such as organising and enhancing personal photos/videos
- Voice recognition
    - Conversion of spoken language into text, identifying and differentiating voices
    - Automation of transcription j*bs
- Image classification
    - Analysis of content of images and organise them into different categories
    - Social media sites to detect and block undesirable content
    - Detect abnormal conditions from medical images
    - Media applications for organising and searching personal photos
- Spam filtering
    - Identify and separate unsolicited and unwanted emails from legitimate emails
    - Protection against phishing attacks, malware and other cyberattacks
- Game playing
    - Serve as opponent or substitute for another human player
    - Researchers may use this technique to explore decision-making strategies in simulations of economic, social and military scenarios
- Content generation
    - [**Generative AI**] - AI that can create new content such as text, images sounds and videos
        - Text: Brainstorming, document summary, reports, writing code
        - Image: Create artwork, design prototypes
        - Sound: Imitate voices, create musical soundtracks

- Video: Produce animations and alter videos to add special effects

5.4.3 Define Machine Learning (ML) as a technique used in AI and explain the difference between ML and traditional programming.

[**Machine learning**] - Technique used in AI that enables a computer to perform complex tasks without explicit programming
[**Model (machine learning)**] - Mathematical representation of the patterns and relationships that are found as a result of training an ML system
[**Training data (machine learning)**] - Collection of examples that is used to teach an ML model how to make predictions or decisions
[**Inference (machine learning)**] - Process of applying an ML model on new data to make predictions or decisions

Aspect	**Machine Learning**	**Traditional Programming**
Approach	Providing a large amount of data for system to identify patterns and implicitly formulate its own rules	Writing explicit rules and instructions for every possible scenario
Flexibility	Adapts to new data and can improve over time with more data	Requires manual modification to change behaviour
Predictability	Depending on type of ML system, output may be probabilistic and vary each time for same input	Output is typically deterministic and repeatable for same input
Speed	Training speed is typically slow; inference speed varies greatly depending on model's size and available resources	Development and execution speeds are typically faster for well-defined problems with clear rules and requirements
Cost	Typically higher cost due to need for specialised hardware (e.g. GPUs) and high power usage	Typically lower cost due to lower hardware and power requirements
Ideal use case	Solving problems with complex patterns difficult or impossible for humans to formulate explicit rules	Solving well-defined problems with clear rules and requirements

5.4.4 Use the nearest neighbour method for a classification task with two quantitative features to demonstrate the basic principles behind ML.

[**Nearest neighbour (NN) method**] - Simple ML algorithm for classification that *assigns to a data point the label of its closest neighbouring data point* in the training data

- Nearest neighbour of a given data point is the one that forms the shortest distance with the given data point (use Pythagoras' Theorem)
- When multiple data points have the same shortest distance:
    - First or last iterated, pick the one that was first/last found
    - Majority voting, pick the one with the most common label

    - Random selection, pick a label randomly

[**Data point (machine learning)**] - A unit of information representing an instance or example relevant to an ML problem
[**Feature (machine learning)**] - A measurable property or characteristic of a data point
[**Label (machine learning)**] - The target output of each data point

5.4.5 Explain how unethical use of AI or using AI with biased data can lead to negative consequences.

[**Overfitting (machine learning)**] - When an ML model learns random noise present in the training data to the extent that it results in poor generalisation when applied to new data
[**Underfitting (machine learning**] - When an ML model is too simple to learn the underlying patterns of the training data
[**Bias (artificial intelligence**] - When an AI system produces results that are systematically prejudiced or discriminatory in an unfair manner

- **Prejudice bias**
    - Training data reflects biases and prejudices that already exist in society, resulting in a trained model producing outputs that perpetuate these biases
- **Sample selection bias**
    - Selection of training data is unbalanced and does not adequately represent the population or the full range of possible data points, resulting in a trained model performing poorly for underrepresented inputs

[**Copyright Ownership and Infringement**]

1. Ownership of AI-Generated Content
    a. Unclear who copyright owner should be for AI-generated content
2. Training on Copyrighted Material
    a. Training data (large amounts) may include copyrighted material collected and/or used without explicit permission of copyright owners
    b. Argument that this may be considered acceptable “fair use” of copyrighted materials for research
3. Generation of Infringing Content
    a. Content may closely resemble copyrighted material even when not prompted to do so explicitly
    b. Depending on similarity, this may constitute plagiarism and/or copyright infringement
        i. Importance of acknowledging use of generative AI as well as to verify and cite sources used by AI to generate the content

[**Spread of Misinformation**]

- [**Deepfake**] - A convincing AI-generated depiction of people doing things they never actually did
- [**Hallucination (artificial intelligence)**] - Falsehoods that are presented as facts by generative AI

[**Safety and Accountability**]

- Safety
    - Usage poses risks resulting in unintended consequences
    - Vulnerability to attacks, prompt injection, where specially crafted malicious inputs can elicit dangerous outputs
    - Difficult to eliminate these risks due to lack of explicit rules or instructions that can be checked thoroughly, along with probabilistic behaviour with varying outputs for the same inputs
- Accountability
    - Unclear as to who should be held accountable for an error, biased decision of unintended consequence due to the use of an AI system without human supervision

[**Changes to the J*b Market**]

- Automation of tasks makes certain j*bs obsolete
    - Great impact for j*bs with repetitive and routine tasks
    - Advent of generative AI may also affect analytical and creative j*bs, which were once thought to be safe from automation
- Upskilling programmes
    - Help workers acquire new skills that will be in demand as j*b market evolves
    - Stuff AI can't do: critical thinking, empathy and interpersonal communication (u still can't find a lover tho HAHA stinky ahh computing+ nerd 🤣🤣🤣🤣🤣🤣👍👍👍👍)
- While AI can automate certain tasks, new j*bs in AI maintenance, supervision and improvement may arise