# Y11 Higher mark written questions

**ALWAYS**:

1. **Read** the question carefully. **Highlight** or underline key words.
2. "**Describe**" questions are about how something works or what happens when… "**Explain**" questions need you to give reasons. If the question says '**why**' always use the word '**because**' in the answer.
3. Start with a **definition** of the main theme. Eg. in a question about artificial intelligence, begin with a description of what artificial intelligence is.
4. Use the **context**. In an exam the question will have a scenario or context. Your answer must refer back to the context in the question.
5. Aim to make 4 **linked points**.

1. What is the Data Protection Act?
2. Explain the ethical and legal issues surrounding the collection of personal data
3. How does computing affect the environment?
4. Explain how having a short replacement cycle for digital devices is harmful for the environment
5. Describe the measures that data centres can take to make them more environmentally friendly
6. What are the causes, impacts & solutions to algorithmic bias?
7. What is an audit trail?
8. What is efficiency in relation to algorithms?
9. Describe how computers represent sound
10. Describe how computers represent bitmap images
11. What are the advantages and disadvantages of different types of compression?
12. Explain the differences between low level and high level programming languages
13. Explain the advantages and disadvantages of high level programming languages
14. Explain the advantages and disadvantages of low level programming languages
15. Describe the function and workings of assembly language
16. Explain the difference between an embedded system and a general purpose computer.
17. Explain the purpose and function of an operating system
18. Explain the difference between compilers and interpreters
19. What are the advantages of using a compiler instead of an interpreter?
20. What are the advantages of using an interpreter instead of a compiler?
21. Explain the ethical issues surrounding the use of artificial intelligence
22. Explain machine-learning
23. What is a robot?
24. Explain the ethical issues surrounding self-driving cars
25. Explain the purpose of a code review
26. Explain technical vulnerabilities in code and the role of software patches and updates
27. Explain poor programming practices and the impacts this has
28. Explain the privacy issues surrounding the use of the internet
29. Explain the issues surrounding the use of lethal autonomous weapons
30. Explain the advantages and disadvantages of social media
31. Explain the differences between a hacker and a penetration tester(ethical hacker)
32. Explain the different types of malware
33. Explain the steps an organisation can take to protect its systems from cyber attacks
34. Explain what a phishing email is
35. What is Intellectual Property and how is it protected?
36. Explain the Copyright, Designs and Patent Act and what it does
37. Explain the role of a Domain Name Service
38. Explain the role of the TCP/IP stack in requesting and receiving webpage
39. Explain the difference between IMAP and POP email protocols.
40. Explain packet-switching and the role of the router
41. Explain the structure of a data packet
42. Explain encryption and its role in securely sending data over the internet
43. Explain the difference between a star and a mesh network
44. Explain how anti malware software works to protect systems from cyber attacks
45. Describe the role of a signature file in anti-malware software
46. Explain the role of a firewall in preventing hacking (unauthorised access)
47. Explain the key features of the Computer Misuse Act.
48. Explain the computational thinking principle of decomposition, with examples.
49. Explain the computational thinking principle of abstraction, with examples.
50. Explain what social engineering is and how organisations and individuals can protect against it
51. Describe baiting, pretexting and quid pro quo cyber attacks.
52. Explain the difference between open source and proprietary software and the advantages and disadvantages of each
53. Explain the disadvantages of open source and proprietary software
54. Explain methods of preventing cyber attacks
55. Explain an Acceptable Use Policy and why an organisation would have one
56. Explain why employees should not use attach their own devices or portable storage to a network
57. Explain the difference between incremental and full back up and why regular backup are vital
58. Explain how to sort data using the bubble sort algorithm
59. Explain how to sort data using the merge sort algorithm
60. Explain how to search data using the linear search algorithm
61. Explain how to search data using the binary search algorithm
62. Compare merge and bubble sort algorithms in terms of efficiency
63. Compare linear and binary search algorithms in terms of efficiency
64. Describe the role of utility software, with examples
65. Explain the Internet of Things and the privacy concerns surrounding it
66. Give some examples of devices connected to the Internet of Things
67. Compare different storage media used for secondary storage
68. Describe the 'stored-program' concept and the Von Neumann architecture
69. Describe the Fetch-Decode-Execute cycle with reference to the buses
70. Explain the different levels of user access rights with examples of each
71. Explain the differences between a Local Area Network (LAN) and a Wide Area Network (WAN)
72. Explain the difference between Wi-Fi and Bluetooth
73. Explain how a ZigBee mesh network works

## What is the Data Protection Act?

The UK Data Protection Act is a law that **protects** the personal information of individuals.

It sets out rules for how organizations must **collect, store, use, and share** personal data, such as name, address, and bank details.

The Act requires organizations to be **transparent** about how they use personal data, and gives individuals the **right to access** their own information.

Organizations must take **appropriate security** measures to keep personal data safe, and must **delete the data** when it is no longer needed.

The Act is enforced by the Information Commissioner's Office (ICO), which can take action against organizations that break the rules.

The Act aims to strike a balance between protecting individuals' **privacy rights** and allowing organizations **to use personal data** for legitimate purposes.

## Explain the ethical and legal issues surrounding the collection of personal data

The collection of **personal data** raises both ethical and legal issues, as it involves the handling of sensitive information about individuals.

From an **ethical** perspective, the collection and use of personal data must be done in a **responsible** and **transparent** manner, ensuring that individuals' **privacy rights** are respected. This includes **obtaining consent** for the collection of data, being clear about **what data is being collected and how it will be used, and protecting the data from unauthorized access or misuse.**

From a legal perspective, the collection and use of personal data is regulated by laws such as the **General Data Protection Regulation (GDPR) and Data Protection Act**. These laws dictate how personal data can be **collected**, **processed**, and **stored**, and also establish the **rights of individuals to access, correct, and delete their personal data**. Organizations must comply with these laws, or they may face fines or other penalties.

Therefore, it is important for organizations to adopt ethical and legal practices when collecting personal data, and to ensure that they have appropriate security measures in place to protect it. This helps to maintain trust with customers and reduces the risk of data breaches or other privacy violations.

## How does computing affect the environment?

Computers and their related technology have both positive and negative impacts on the environment.

**Positive impacts include:**

- Improved energy efficiency, reducing energy waste and carbon emissions
- Facilitating telecommuting and remote work, reducing transportation-related emissions
- Helping to track and manage environmental data and resources more effectively such as Environmental monitoring devices, such as air quality sensors and water quality monitoring systems

**Negative impacts include:**

- Large amounts of electronic waste, which can harm wildlife and the environment if not disposed of properly
- Energy consumption, which contributes to greenhouse gas emissions and global warming
- The extraction of raw materials for computer production, which can lead to habitat destruction and pollution

Overall, the impact of computers on the environment is complex and multifaceted. To minimize the negative impacts and maximize the positive, it's important to use technology responsibly and take steps to reduce waste and emissions.

## Explain how having a short replacement cycle for digital devices is harmful for the environment

**A short replacement cycle** for digital devices refers to the practice of frequently replacing electronics such as smartphones, computers, and other gadgets with newer models. This practice contributes to **electronic waste (e-waste)** and has a harmful impact on the environment for several reasons:

**Resource Depletion**: The production of new electronic devices requires large amounts of resources, including metals, minerals, and rare earth elements. These resources are finite and are being depleted at a rapid rate due to the increasing demand for electronics.

**Energy Consumption**: The production of new electronic devices requires large amounts of energy, which contributes to greenhouse gas emissions and other forms of environmental pollution.

**E-waste**: When electronic devices are discarded, they often end up in landfills where toxic chemicals and metals can leach into the soil and water, causing harm to the environment and human health.

**In conclusion**, having a short replacement cycle for digital devices contributes to resource depletion, energy consumption, and e-waste, all of which are harmful to the environment. To reduce the environmental impact of electronics, it is important to adopt more sustainable practices, such as using devices for longer periods, repairing and upgrading devices, and properly disposing of e-waste.

## Describe the measures that data centres can take to make them more environmentally friendly

**Data centres** are large facilities that house servers and other computing equipment that store and process large amounts of data. To make them more environmentally friendly, data centres can take the following measures:

**Energy Efficiency**: Data centres can reduce energy consumption by using **energy-efficient hardware**, such as servers and storage devices, and implementing energy-saving measures, such as **turning off equipment** when not in use. They can also use **renewable energy sources**, such as solar or wind power, to reduce their carbon footprint.

**Cooling Efficiency**: Data centres can reduce cooling energy consumption by using energy-efficient cooling systems, such as using outside air or using cooling systems that use less energy. Siting data centres in cool climates reduces the need for energy-hungry air conditioning units as air can be brought in from outside and circulated.

R**ecycling and Reuse**: Data centres can recycle and reuse components, such as servers, storage devices, and other equipment, to reduce e-waste and conserve resources.

By implementing these measures, data centres can reduce their energy consumption, conserve resources, reduce e-waste, and minimize their impact on the environment.

## What are the causes, impacts & solutions to algorithmic bias?

Algorithmic bias refers to the systematic error in a computer algorithm that leads to unequal or unfair treatment of certain groups or individuals.

**Causes of algorithmic bias include:**

The use of biased training data to train the algorithm, which can perpetuate and amplify existing biases in society

The design choices made by the developers, who may unconsciously bring their own biases into the algorithm

The lack of diversity in the development team, which can limit the perspectives and experiences taken into account when building the algorithm

**Impacts of algorithmic bias can include:**

Discriminatory treatment of certain groups or individuals, such as in hiring, lending, or criminal justice

The reinforcement of stereotypes and inequality in society

The loss of public trust in technology and its applications

It's important to recognize and address algorithmic bias in order to ensure that technology is used in a fair and just manner. This can be done by using diverse and unbiased training data, promoting diversity in the development team, and regularly auditing and testing algorithms for biases.

## What is an audit trail?

An audit trail, also known as an audit log, is a record of every activity or event that takes place on a computer system or network.

**The purpose of an audit trail** is to provide a chronological record of actions taken on a system, so that administrators and auditors can later review and understand what has happened. For example, an audit trail can be used to:

- Track who made changes to sensitive data, when the changes were made, and what was changed
- Monitor and detect potential security breaches or unauthorized access to a system
- Investigate errors or malfunctions in a system

**An audit trail typically contains** information such as the date and time of the event, the user who performed the action, the type of action taken, and any relevant details or parameters associated with the action. The audit trail can be stored in a separate database, a log file, or any other type of persistent storage.

In simple terms, an audit trail is a history of what has happened on a computer system, which can be used to help ensure the security, reliability, and accountability of the system.

## What is efficiency in relation to algorithms?

**Algorithmic efficiency** refers to how well a computer algorithm performs in terms of using resources like time and memory.

When designing an algorithm, it's important to consider its efficiency because a **fast** and efficient algorithm will run quickly and smoothly, while a slow and inefficient algorithm may take a long time to complete or use too much **memory** and cause the system to slow down or crash.

The efficiency of an algorithm is usually measured in terms of time complexity, which refers to the **amount of time** an algorithm takes to complete its task, and space complexity, which refers to the **amount of memory** or storage an algorithm uses.

**For example,** imagine you have two algorithms for sorting a list of numbers. Algorithm A takes 10 seconds to sort a list of 1000 numbers, while algorithm B takes 5 seconds to sort the same list. Algorithm B is more efficient in terms of time complexity, because it sorts the numbers faster.

**In simple terms**, algorithmic efficiency refers to how quickly and effectively a computer algorithm uses resources to complete its task. The goal of efficient algorithms is to complete tasks as quickly and efficiently as possible, with the least amount of resources.

## Describe how computers represent sound

Computers represent sound as a series of numbers, known **as digital audio data**. These numbers represent the **amplitude**, or volume, of sound waves at **regular intervals of time**.

When a sound is recorded, it is **sampled** many times per second and each sample is assigned a numerical value. For example, a commonly used sample rate is 44,100 samples per second, which is commonly referred to as 44.1 **kHz**. This means that the sound is measured 44,100 times every second and each measurement is assigned a numerical value.

The numerical values are then stored in a digital file, such as a WAV or MP3 file. When the file is played back, the numbers are used to generate an electrical signal that is converted into sound waves by a speaker.

In simple terms, computers represent sound as a series of numbers, which describe the volume of sound at regular intervals of time. These numbers can be stored and played back as digital audio.

## Describe how computers represent bitmap images

A bitmap image is a type of digital image that represents pictures as a grid of tiny **pixels**, each with a specific colour. To represent a **bitmap image** on a computer, each pixel is assigned a binary value that determines its colour. The colour of each pixel is determined by a binary code, which can be represented as a combination of 1s and 0s.

**Colour depth** and **resolution** are two important aspects of digital images that affect their quality and appearance.

**Colour depth** refers to the number of bits of information used to represent the colour of each pixel in a digital image. The higher the colour depth, the more colours can be represented, and the more vivid and accurate the image will appear. For example, a digital image with 8-bit colour depth can represent 256 colours, while a digital image with 16-bit colour depth can represent 65,536 colours.

**Resolution** refers to the number of **pixels per inch** in an image, and it affects the level of detail and sharpness of an image.

A digital image with high colour depth and resolution will look much better than an image with low colour depth and resolution.

## What are the advantages and disadvantages of different types of compression?

**Lossless and lossy Compression** are two methods used to reduce the size of a file while retaining (or losing) as much data as possible.

**Lossless Compression: Advantages:**

- Retains all data: The compressed file can be restored to its original form without any loss of data.
- Ideal for file types where data loss is unacceptable, such as text files, images, etc.

**Disadvantages**:

- Does not reduce file size as much as lossy compression.
- Typically requires more processing power to compress and decompress files.

**Lossy Compression: Advantages:**

- Compresses files to a much smaller size compared to lossless compression.
- Ideal for file types where some data loss is acceptable, such as audio, video.

**Disadvantages:**

- Data is lost: The compressed file cannot be restored to its original form as some data has been lost during the compression process.
- Quality may be affected: Depending on the amount of compression applied, the quality of the file may be significantly impacted.

**In summary,** lossless compression is typically used for file types where data loss is unacceptable, such as text and some image files. Lossy compression is used for file types where some data loss is acceptable, such as audio and video files, in order to achieve smaller file sizes.

## Explain the differences between low level and high level programming languages

High-level and low-level programming languages are different based on the amount of abstraction they provide from the underlying computer hardware.

**High-level programming languages**, such as Python are designed to be **easy to read, write and learn**, and they hide many of the complexities of the computer's architecture. These languages are often used for building applications that require complex algorithms and data structures, and they are commonly used in areas such as web development, data analysis, and artificial intelligence.

**Low-level programming languages, such as Assembly Language**, provide closer access to the computer's hardware and are designed for systems-level programming. These languages are often used for writing system software, such as operating systems, **embedded systems** and **device drivers**.

Low-level programming languages are also used for **real-time embedded systems**, as they allow for precise control over the computer's hardware.

**Low-level programming languages** are often used in situations where close access to the computer's hardware and maximum control over the system's resources are required**. Here are some examples** of when a low-level programming language is used:

**Device Drivers**: Device drivers, which allow the operating system to communicate with specific hardware devices, are typically written in low-level programming languages.

**Embedded Systems**: Embedded systems, such as those found in medical devices, cars, and mobile phones, often use low-level programming languages because these systems require maximum control over hardware resources and minimum overhead.

**In summary**, high-level programming languages are used for application-level programming and are designed to be easy to read and write, while low-level programming languages are used for system-level programming and provide closer access to the computer's hardware.

## Explain the advantages and disadvantages of high level programming languages

High-level programming languages and low-level programming languages are two different categories of programming languages, and each has its own advantages and disadvantages.

**Advantages of high-level programming languages:**

- **Easier to learn and use**: High-level programming languages are designed to be more user-friendly, with a simpler syntax and more abstract concepts. This makes them easier to learn and use, especially for beginners.
  **More abstract**: High-level programming languages abstract away many of the details of the underlying hardware, allowing you to focus on the logic and functionality of your program, rather than the details of memory management and other low-level tasks.
  **Faster development time**: With high-level programming languages, you can write code more quickly and with fewer errors, because the language provides higher-level abstractions and built-in functions for common tasks.

**Disadvantages of high-level programming languages:**

- **Slower execution speed**: High-level programming languages can be slower than low-level programming languages because the abstractions and built-in functions add overhead to the execution of the program.
  **Less control over the hardware**: High-level programming languages provide less direct control over the underlying hardware, which can limit the types of programs you can write and their performance.
  **Resource-intensive**: High-level programming languages require more memory and processing power to run the same program.

## Explain the advantages and disadvantages of low level programming languages

**Advantages of low-level programming languages:**

- **Faster execution speed:** Low-level programming languages provide closer control over the underlying hardware, so programs written in these languages can be faster and more efficient than those written in high-level programming languages.
  **Better control over the hardware**: Low-level programming languages allow you to access and control the hardware at a much deeper level, which can be useful for tasks like device drivers, firmware, and operating systems.
  **More efficient use of resources:** Programs written in low-level programming languages can be more memory- and processing power-efficient than those written in high-level programming languages.

**Disadvantages of low-level programming languages:**

- Har**der to learn and use:** Low-level programming languages such as Assembly Language are generally more difficult to learn and use than high-level programming languages, because they require a deeper understanding of the underlying hardware and a more complex syntax.
  **More error-prone:** Low-level programming languages can be more error-prone than high-level programming languages, because small mistakes can have big consequences in the underlying hardware.
  **Slower development time**: Programs written in low-level programming languages can take longer to write, debug, and maintain than those written in high-level programming languages.

# Describe the function and workings of assembly language

Assembly language is a low-level programming language that is used to write programs for computers. It provides a way to write programs that are closer to the machine language used by the computer's hardware, which allows for more fine-tuned control over the computer's operation.

**Mnemonics** are a type of shorthand used in assembly language programming to make it easier for the programmer to write and remember the code. They are short, memorable words or phrases that represent the machine code instructions that the processor will execute. For example, the instruction to load a value into a register might be represented by the mnemonic "LD" followed by the register number and the value to be loaded.

Using **mnemonics** makes the code more readable and easier to write, which in turn makes it easier to debug and maintain.

**Instruction Set:** Assembly language consists of a set of instructions that the computer can understand and execute. Each instruction represents a single operation that the computer can perform, such as moving data from one place to another, performing arithmetic operations, or jumping to a different part of the program.

**Machine-Specific**: Assembly language is machine-specific, which means that the instruction set and the format of the instructions will vary depending on the type of computer for which the program is being written.

**Register Access:** Assembly language provides direct access to the computer's hardware, such as the CPU's registers, which are small areas of memory that the CPU uses to store data and perform operations. This direct access allows assembly language programs to run faster and use the hardware more efficiently than higher-level languages.

**Assembly Process**: Assembly language programs are written in text files and then assembled into machine language by an **assembler**. The assembled machine language program can then be executed by the computer's hardware.

**Limitations:** Assembly language can be difficult to read and write, as it requires a deep understanding of the computer's hardware and the instruction set. Assembly language programs are also harder to maintain and debug than programs written in higher-level languages.

**Assembly language** is used in a variety of applications, including operating systems, device drivers, firmware, and other low-level software. Despite its limitations, it remains an important tool for low-level programming tasks and for writing efficient, high-performance code.

# Explain the difference between an embedded system and a general purpose computer.

A **general purpose computer** and an **embedded system** are both computers, but they have different design goals and are used for different purposes.

A **general purpose computer** is a computer that can be used for many **different** tasks, such as browsing the web, running office software, playing games, and more. These computers are often designed to be **flexible** and have a lot of resources, such as **processing power, memory, and storage**.

An **embedded system**, on the other hand, is a computer system that is integrated into a device to perform a specific task. It is designed to perform **a single or a limited number of functions** Embedded systems are designed to be **compact, efficient, and dedicated to a single task**. They typically have **limited memory and use less power** and have lower processing powers.

**Examples of embedded systems include:**
**Automobiles**: Cars have embedded systems that control the engine, transmission, and other systems.
**Television sets**: TV sets have embedded systems that control the display and sound.
**Mobile phones**: Mobile phones have embedded systems that control the display, sound, and communication functions.
**Home appliances**: Many home appliances such as washing machines, refrigerators, and microwave ovens have embedded systems that control their functions.

**Medical equipment**: Medical equipment such as heart monitors, blood glucose meters, and X-ray machines have embedded systems that control their functions.

## Explain the purpose and function of an operating system

An operating system (OS) is the software that **manages** and **controls** the **hardware** and other **software** on a computer. It acts as an **interface** between the computer's hardware and the programs you run on your computer.

The main purpose of an operating system is to manage the resources of the computer, such as **memory, processing power, and input/output operations**. It is responsible for tasks such as **managing files and directories**, **managing security**, **managing memory, managing users and managing peripheral devices**, like printers, keyboards, and mice.

The operating system also serves as a platform for other software to run on. For example, you can run a web browser or a text editor on top of the operating system. This allows the hardware to be used in a flexible way, so that different types of tasks can be performed.

Overall, the operating system plays a crucial role in making a computer useful and efficient for users.

## Explain the difference between compilers and interpreters

**Compilers and interpreters** are two different approaches to translating code written in a high-level programming language into machine code that a computer can understand and execute. Both are **translators**.

A **compiler** is a program that reads source code and converts it into an executable form, typically machine code for a specific computer architecture. Once the code has been compiled, the machine code can be executed directly without the need for a compiler to be present. This process is usually performed before the code is executed and generates a standalone executable file that can be run without any dependencies.

An **interpreter**, on the other hand, does not produce an executable file. Instead, it directly executes the source code by translating it line by line into machine code and executing it on the fly. This process eliminates the need for a separate compile step, but it also typically results in slower performance compared to compiled code because the interpreter has to repeatedly perform the translation process.

**In summary**, compilers convert source code into machine code and create an executable file, while interpreters execute source code directly and do not produce an executable file.

## What are the advantages of using a compiler instead of an interpreter?

- The main advantage of using a **compiler** instead of an **interpreter** is improved **performance**. Since the code is translated into machine code before it is executed, compiled programs **run faster** than interpreted programs, which need to be translated every time they are run.
- Another advantage of using a compiler is that compiled programs are typically **more secure** than interpreted programs, as the source code is not visible once it has been compiled into machine code. This makes it more difficult for attackers to reverse engineer the code and find vulnerabilities.
- In summary, the main advantages of using a compiler instead of an interpreter are improved performance and increased security.

## What are the advantages of using an interpreter instead of a compiler?

The main **advantage** of using an **interpreter** instead of a compiler is that it provides greater flexibility and ease of use. Interpreters can execute code **as soon as it is written**, without the need for a separate compilation step, which makes it **easier to test and debug code**.

Another **advantage** of using an interpreter is that it can be more **platform-independent**, as interpreted code does not need to be recompiled for different architectures or operating systems. Interpreters handle errors immediately, as they will **stop executing at the point of an error**, allowing the programmer to fix the issue and continue from where they left off.

## Explain the ethical issues surrounding the use of artificial intelligence

- **Artificial intelligence (AI)** is a rapidly advancing field with the potential to revolutionize many aspects of our lives, but it also raises a number of ethical concerns. Here are some of the most significant ethical issues surrounding the use of AI:
- **Bias and discrimination:** AI systems are often trained on large datasets, which can reflect the biases and prejudices of the people who created them. This can result in discriminatory outcomes, such as biased hiring decisions or unequal treatment of people based on their race, gender, or other characteristics.
- **Privacy**: AI systems often require access to vast amounts of personal data, which can pose a threat to privacy. There are concerns that AI systems could be used to monitor or manipulate individuals, or that the data collected by AI systems could be misused.
- **Job displacement**: AI systems are increasingly being used to automate tasks previously performed by humans, which could result in job losses and increased inequality. There are concerns that this could lead to widespread unemployment and further exacerbating economic and social problems.
- **Responsibility and accountability**: AI systems can make decisions that have significant impacts on people's lives, but it can be difficult to determine who is responsible when things go wrong. This raises questions about accountability and the need for regulations to ensure that AI systems are used in an ethical and responsible manner.
- **Autonomous systems:** Some AI systems, such as autonomous vehicles or military drones, are capable of making decisions and taking actions without human intervention. This raises concerns about safety and the need for ethical guidelines to ensure that these systems are used in a responsible and predictable manner.
- **In summary,** the use of AI raises a number of ethical issues, including bias and discrimination, privacy, job displacement, responsibility and accountability, and autonomous systems. It is important for society to consider and address these issues as AI continues to advance and become more prevalent in our lives.

## Explain machine-learning

- Machine learning is a method of teaching computers to make decisions and predictions based on patterns and relationships in data, without explicitly programming them. In other words, it's a way for computers to automatically improve their performance on a task, by learning from experience. The machine analyses large amounts of data, finds patterns and connections in the data, and uses this information to make predictions or decisions, without being explicitly told how to perform the task.
- The ethical issues are the same as explained for AI

## What is a robot?

**A robot** is a machine designed to carry out a complex series of tasks **automatically**, especially by being programmed by a computer. Robots can be used in a variety of applications, including manufacturing, healthcare, and military.

**Ethical issues** are the same as for any AI. Eg**. Job displacement**: The widespread use of robots has the potential to automate many jobs, which could lead to widespread job displacement and unemployment.

**Responsibility and accountability**: Robots can make decisions and take actions that have significant impacts on people's lives, but it can be unclear who is responsible and accountable for their actions.

**Privacy concerns**: Robots often use and collect large amounts of personal data, which can raise privacy concerns and the risk of data breaches.

**Bias and discrimination**: Robots can perpetuate and even amplify existing biases in society if the algorithms used to control them contain such biases. This can lead to discriminatory outcomes in areas such as hiring, lending, and policing.

**Lack of transparency**: Some robots can be difficult to understand and interpret, making it challenging to assess their decision-making processes and identify potential biases.

These ethical issues highlight the need for responsible development, deployment, and regulation of robots, to ensure that they are used in ways that benefit society and do not perpetuate harmful biases or discrimination.

## Explain the ethical issues surrounding self-driving cars

**Self-driving cars** are vehicles equipped with advanced technologies that allow them to drive autonomously, without the need for a human driver. The ethical issues surrounding self-driving cars include:

**Responsibility and accountability**: In the event of an accident, it can be unclear who is responsible and accountable for the actions of a self-driving car.

**Safety**: Self-driving cars raise questions about the safety of passengers, pedestrians, and other road users. The technology is still developing, and there are concerns about the reliability and accuracy of the sensors and algorithms used in self-driving cars.

**Bias and discrimination:** Self-driving cars can perpetuate and even amplify existing biases in society if the algorithms used to control them contain such biases. This can lead to discriminatory outcomes in areas such as route selection, emergency braking, and collision avoidance.

**Job displacement:** Self-driving cars have the potential to automate many jobs in the transportation sector, which could lead to widespread job displacement and unemployment.

**Privacy concerns:** Self-driving cars collect and use large amounts of personal data, which can raise privacy concerns and the risk of data breaches.

These ethical issues highlight the need for responsible development, deployment, and regulation of self-driving cars, to ensure that they are used in ways that benefit society and do not cause harm.

## Explain the purpose of a code review

A **code review** is the process of systematically reviewing and analysing the source code of a software program to **identify potential issues**, **improve the quality of the code**, and **maintain a high level of software development standards.** The **purpose** of a code review is to:

**Improve code quality:** Code reviews help identify and correct errors, bugs, and other issues in the code that might impact the functionality and performance of the software.

**Ensure code standards:** Code reviews help ensure that the code follows established coding standards, such as naming conventions, coding styles, and best practices.

**Share knowledge:** Code reviews provide an opportunity for developers to share their knowledge and expertise with each other, and to learn from each other's experiences.

**Identify potential security issues:** Code reviews can help identify potential security risks and vulnerabilities in the code, and help ensure that the software is secure and meets security standards.

Overall, code reviews are an important part of the software development process, and help to ensure the quality, security, and maintainability of the code.

## Explain technical vulnerabilities in code and the role of software patches and updates

- **Technical vulnerabilities** in code are **weaknesses** or flaws in the software that can be **exploited by attackers** to gain **unauthorized access** to a system or steal sensitive information. These vulnerabilities can arise from various causes, such as **coding errors, poor security design, or the use of outdated technologies.**
- **Software patches and updates** play an important role in addressing technical vulnerabilities in code. A software **patch** is a small piece of code that **fixes a specific problem or vulnerability** in the software. A software **update** is a more comprehensive change that may include multiple patches and new features.
- By installing software patches and updates, users can close the security holes and **prevent attackers from exploiting vulnerabilities**. This helps to keep their systems and data secure, and reduces the risk of cyber attacks.

## Explain poor programming practices and the impacts this has

**Poor programming practices** refer to habits and techniques in software development that result in **low-quality, unreliable, and potentially insecure code.** Some common examples of poor programming practices include:

**Hard-coding values instead of using variables and constants**: This makes the code less flexible and harder to maintain.

**Failing to validate user inputs**: This can lead to security vulnerabilities.

**Not handling errors and exceptions properly**: This can cause the program to crash or produce incorrect results.

**Ignoring code maintainability and readability**: This means not including comments or using white space to separate chinks of code. This makes it harder for other developers to understand and modify the code.

**The impacts of poor programming practices** can be significant, including:

**Increased costs and delays** due to bugs and vulnerabilities: Fixing poor code takes longer and costs more than fixing high-quality code.

**Decreased reliability and security**: Poorly written code is more prone to errors and security vulnerabilities, putting the user data and systems at risk.

**Loss of business and reputation**: Poor code can cause the software to fail in important ways, leading to customer frustration and loss of business.

It is important for software developers to follow best practices and maintain high standards for coding quality to ensure that their code is reliable, secure, and maintainable.

## Explain the privacy issues surrounding the use of the internet

The privacy issues surrounding the use of the internet include:

**Data collection and usage**: Companies and organizations collect and store large amounts of personal data about users, including their browsing history, search queries, and demographic information. This data is often used for targeted advertising, which can raise privacy concerns.

**Data breaches**: The storage of large amounts of personal data by companies and organizations also increases the risk of data breaches, in which sensitive information is lost or stolen.

**Online tracking**: Companies and organizations can track users' online behaviour to build profiles of their interests and habits, which can be used for targeted advertising or shared with third parties.

**Government surveillance**: Governments around the world have been known to monitor and collect data from internet users, which can raise privacy concerns and limit freedom of expression.

**Cybercrime**: The internet is also a prime target for cybercriminals, who use tactics such as phishing, malware, and ransomware to steal personal information or hold data hostage.

**To protect their privacy,** users can take steps such as using strong **passwords**, enabling **two-factor authentication**, using **encrypted** communication tools, and being cautious about the information they share online. However, to fully address the privacy issues surrounding the use of the internet, a more comprehensive approach that includes stronger privacy laws and regulations is needed.

## Explain the issues surrounding the use of lethal autonomous weapons

**Lethal autonomous weapons** (LAWs) are weapons systems that are designed to operate **without human** intervention and to make decisions about the use of deadly force. The issues surrounding the use of these weapons include:

**Responsibility and accountability:** The deployment of LAWs raises questions about who is responsible and accountable for their actions in the event of unintended harm, injury, or death.

**Lack of human control:** The lack of human control in the decision-making process raises concerns about the ethical and moral implications of delegating the power to take life to machines.

**Bias and discrimination**: There are concerns that the algorithms used to control LAWs could perpetuate existing biases and discrimination, leading to disproportionate harm to certain populations.

**Unintended consequences**: The deployment of LAWs could have unintended consequences, such as escalation of violence, increased civilian casualties, or destabilization of regional security.

**Legal and ethical challenges**: There are also legal and ethical challenges associated with the deployment of LAWs, including questions about the legitimacy of their use under international humanitarian law, human rights law, and other relevant legal frameworks.

These issues highlight the need for caution and careful consideration in the development and deployment of LAWs, and for the establishment of clear norms, regulations, and international legal frameworks to govern their use.

## Explain the advantages and disadvantages of social media

**Advantages of social media:**

**Connectivity**: Social media allows people to connect and communicate with each other, regardless of geographical barriers.

**Information Sharing:** Social media provides an easy and fast way to share information, news, and ideas with a large audience.

**Increased Exposure**: Social media can help individuals and businesses gain more exposure and attract more customers.

**Improved Marketing:** Social media provides businesses with a cost-effective way to reach their target audience and promote their products or services.

**Networking**: Social media can help people build professional networks and make new connections.

**Disadvantages of social media:**

**Addiction**: Some people can become addicted to social media and spend too much time using it, neglecting other important aspects of their lives.

**Cyberbullying:** Social media can provide a platform for cyberbullying, harassment, and hate speech, which can have negative effects on individuals, especially children and teenagers.

**Misinformation**: Social media can spread false or misleading information quickly and easily, which can cause confusion and harm.

**Decreased Privacy:** Social media can compromise privacy by allowing users to share too much personal information or by collecting and sharing data without consent.

**Mental Health:** The constant exposure to comparison and perfection on social media can lead to negative self-esteem, anxiety, and depression.

## Explain the differences between a hacker and a penetration tester(ethical hacker)

- A **hacker** is someone who uses their technical skills to gain **unauthorized access to computer systems or networks with malicious intent**. They often do this for **personal gain**, such as stealing sensitive information or causing harm to the system or its users.
- On the other hand, a **penetration tester** is a professional who is hired by organizations to legally and ethically test the security of their computer systems and networks. The **goal of a penetration tester** is to identify any vulnerabilities or weaknesses in the system and provide recommendations on how to improve security. **Penetration testing** is performed with the permission of the organization and is an important part of ensuring the overall security of a system.
- **In simple terms**, a hacker is someone who breaks into systems illegally and with harmful intent, while a penetration tester is someone who breaks into systems legally and with the goal of improving security.

## Explain the different types of malware

**Malware is short for malicious software**, which refers to any software **designed to harm** or exploit a computer system. There are several types of malware, including:

**Virus**: a type of malware that **replicates** itself by inserting its code into other files or programs, **spreading** from one computer to another.

**Trojan**: a type of malware that **disguises itself** as a legitimate program, but once installed, it gives attackers access to the computer system, allowing them to steal data, install additional malware, or cause other harm.

**Worm**: a type of malware that **spreads** itself across networks and computer systems without the need for a host file or program.

**Ransomware**: a type of malware that **encrypts** a victim's files and **demands a ransom payment** to restore access.

**Adware**: a type of malware that **displays unwanted or intrusive advertisements** on the infected computer.

**Spyware**: a type of malware that collects information about a victim's computer usage and sends it to the attacker, often without the victim's knowledge or consent.

It's important to take the necessary steps to protect your computer from malware, such as keeping your software up to date, using antivirus software, and being cautious when opening email attachments or clicking on links from unknown sources.

## Explain the steps an organisation can take to protect its systems from cyber attacks

**Cyber attacks** can cause significant harm to an organization, so it's important to take steps to protect your systems. Here are some simple steps that organizations can take to reduce their risk:

- Use strong **passwords** and update them regularly.
- Keep software and systems **up to date** with the latest security patches.
- Use **antivirus so**ftware and keep it updated.
- **Enable firewalls** to prevent unauthorized access to your network.
- **Train employees** on safe computing practices, such as avoiding suspicious emails and websites, and not sharing sensitive information.
- **Regularly backup** important data to prevent data loss in case of an attack.
- **Limit the amount of sensitive information** stored on systems and encrypt sensitive data when stored.
- **Implement access controls** to limit who can access sensitive information and systems.
- **Regularly monitor systems** and networks for signs of intrusion or unusual activity.

By taking these steps, organizations can reduce their risk of a cyber attack and better protect their systems and sensitive information.

## Explain what a phishing email is

- A phishing email is a type of **social engineering.** It is a scam email that aims to **trick the recipient** into revealing sensitive information, such as passwords, credit card numbers, or other personal information. These emails often appear to come from a reputable source, such as a bank, a well-known company, or even a friend, and they often include **a sense of urgency or a false threat** to pressure the recipient into taking action.
- For example, a phishing email might claim that your bank account is about to be suspended and **ask you to click on a link** to update your information. If you do, you'll be taken to a fake website that looks like your bank's site and prompted to enter sensitive information, which is then collected by the attacker.
- It's important to be vigilant when it comes to emails that ask for sensitive information and to never enter personal information on a website unless you are sure it is legitimate. If you receive an email that you suspect is a phishing attempt, you should delete it immediately and **not click on any links or download any attachments.**

## What is Intellectual Property and how is it protected?

**Intellectual property** refers to creations of the mind, such as inventions, literary and artistic works, symbols, and designs, that are used in business. It includes images, music, and written work including software/apps

Intellectual property is protected through a combination of legal rights and laws that give creators, inventors, and owners exclusive rights to control the use of their creations for a certain period of time. This allows them to earn recognition and financial benefit from their work.

**Trademarks**: A trademark is a symbol, word, or phrase used to identify a brand and distinguish it from others. Trademarks are protected by trademark law and can be registered with the government.

**Copyrights**: A copyright is a legal right that gives the creator of an original work, such as a book, song, or movie, exclusive control over the use and distribution of that work. Copyrights are automatically granted as soon as a work is created and can be registered with the government for additional protection.

**Patents**: A patent is a legal right that gives the inventor of a new and useful product or process exclusive control over the use and manufacture of that invention for a certain period of time. Patents are granted by the government and give the inventor a monopoly on their invention for a limited time.

## Explain the Copyright, Designs and Patent Act and what it does

The **Copyright, Designs and Patents Act (CDPA) is** a UK law that governs the protection of **intellectual property**, including copyrights, designs, and patents. The CDPA provides the legal framework for protecting creative works, such as books, music, paintings, and photographs, as well as inventions and designs.

The CDPA gives creators and owners of intellectual property the exclusive **right** to control the use of their works and to **prevent others from copying, distributing, or otherwise using their works without permission**. This includes the right to make copies, sell or license their works, and to earn money from them.

The CDPA also provides legal remedies for individuals and organizations whose intellectual property rights have been infringed, such as the **right to take legal action** to stop the infringement and to recover damages.

Overall, the CDPA is designed to promote creativity and innovation by protecting the rights of creators and owners of intellectual property, while also ensuring that the public has access to a wide range of creative works and inventions.

## Explain the role of a Domain Name Service

A Domain Name System (DNS) is a critical component of the internet that helps **translate human-friendly domain names**, such as www.example.com, into the **numerical IP addresses** that computers use to communicate with each other. The DNS acts as a kind of directory, allowing computers to locate the correct IP address for a given domain name.

**Here's how it works**: when a user types a domain name into their **browser**, the browser sends a **GET request** to the **DNS server** to resolve the domain name into an IP address. The DNS server then **looks up the IP address** associated with the domain name in its database and returns the information to the user's browser. The browser then uses the IP address to establish a connection with the website's server and retrieve the requested web page.

The role of the **DNS is essential because** it makes it possible for users to easily access websites and other online resources using simple, easy-to-remember domain names, rather than having to memorize the numerical IP addresses of each resource. This makes the internet more user-friendly and accessible to a wider range of users.

Overall, the DNS is a critical component of the internet infrastructure that helps ensure that users can find and access the online resources they need, quickly and easily.

## Explain the role of the TCP/IP stack in requesting and receiving webpage

The **Transmission Control Protocol/Internet Protocol (TCP/IP) stack** is the standard set of communication protocols used for transmitting data over the internet. When a user requests a webpage, the TCP/IP stack plays a crucial role in facilitating the exchange of information between the user's device and the server hosting the webpage.

**Here's how it works:**

The user's device sends a request for a webpage using **the Hypertext Transfer Protocol (HTTP**).This happens on the **Application layer**.
The request is divided into **small packets of data** on the **Transport layer** using TCP.
The **Internet layer** is responsible for routing the packets to their destination by forwarding them through various network devices, such **as routers using IP**.
The **Link Layer** is responsible for transmitting raw data bits over a physical medium. The protocols in this layer are **Ethernet** and **Wi-Fi**
**At the destination,** the **TCP layer** reassembles the packets and checks to ensure that they have been received correctly. If any packets are missing or corrupted, the TCP layer requests that they be resent.
The **server** hosting the webpage **receives the request, processes it, and sends back the requested webpage in the form of HTML code.**
The HTML code is divided into **packets** and sent back to the user's device **using the IP layer**.

The **TCP layer on the user's device reassembles the packets and checks for errors.** If everything is in order, the HTML code is passed to the browser for rendering and display.

## Explain the difference between IMAP and POP email protocols.

- **IMAP** (Internet Message Access Protocol) and **POP** (Post Office Protocol) are two different **protocols** used for retrieving email from a mail server.
- **POP** is a simple, older protocol that downloads email from the server to a **single device**, such as a computer or a smartphone. Once the email is downloaded, it is **deleted from the email server** and it is only stored on the device and can only be accessed from that device.
- **IMAP**, on the other hand, allows email to be **stored and managed on the server**, rather than on a single device. This means that email can be accessed from **multiple devices**, such as a computer, a smartphone, and a tablet, and changes made on one device will be reflected on all devices.
- **IMAP** also allows users to keep their email organized on the server, by creating folders and storing emails in different folders. This makes it easier to manage and find emails, especially for users who have a large number of emails.
- **In summary**, the main difference between IMAP and POP is that IMAP allows email to be stored and managed on the server, while POP downloads email to a single device. IMAP is generally considered to be a more sophisticated and flexible protocol, while POP is simpler and more suited to users who need to access email from a single device.

## Explain packet-switching and the role of the router

**Packet switching** is a method of transmitting data over a network by breaking it into smaller units, called **packets**, and sending each packet individually to its **destination**. The packets are **reassembled** at the destination to form the original message.

Each **packet contains information about its source, destination, and the order in which it should be reassembled, as well as the data itself**. When a packet is transmitted over the network, it may take a **different path** from other packets, as the network **routes each packet based on the current network conditions**. This makes packet switching more **efficient** than other methods, as the network can use its resources more effectively.

**The role of a router** in a packet-switched network is to determine **the best path** for each packet to take to its destination, based **on network conditions**, and to **forward the packets** to their destinations. Routers use **routing tables,** which are lists of routes and their corresponding costs, to determine the best path for each packet.

When a packet arrives at a router, **the router examines the destination address in the packet header**, and looks up the corresponding route in its routing table. If the destination is not directly connected to the router, the router forwards the packet to the next hop along the best path, until the packet reaches its final destination.

**In summary,** packet switching is a method of transmitting data by breaking it into smaller units and sending each unit individually to its destination. Routers play a crucial role in packet-switched networks by determining the best path for each packet to take to its destination and forwarding the packets to their destinations.

## Explain the structure of a data packet

- A **data packet** is a unit of data that is transmitted over a network and contains information that is necessary for its successful delivery. The structure of a data packet can be broken down into several parts:
- **Header**: The header is the first part of the data packet and contains information about the packet itself, **such as the source and destination addresses, the length of the packet, and the type of data contained in the packet.**
- **Body**: The body of the packet contains the **actual data** that is being transmitted. It can be anything from text, images, audio, or video.
- **Footer**: The footer is the final part of the data packet and contains **error-checking information, such as a checksum** to ensure the data has not been corrupted during sending.
- The header, body, and footer are combined to form a single data packet, which is then transmitted over a network and reassembled at the destination. The structure of the data packet is important because it ensures that the data is transmitted efficiently, securely, and accurately.

## Explain encryption and its role in securely sending data over the internet

**Encryption** is the process of converting plain text into a scrambled form to prevent unauthorized access to the information. When data is **encrypted**, it can only be read by someone with the appropriate **key** or password to **decrypt** it.

**The role of encryption** in securely sending data over the internet is to **protect the privacy** and confidentiality of the information being transmitted. When **sensitive information**, such as credit card numbers or passwords, is transmitted over the internet, it is vulnerable to interception by unauthorized third parties. **Encryption** helps to ensure that even if the data is intercepted, **it cannot be read or understood** without the decryption key.

**Encryption** is used in a number of different applications, including email, virtual private networks (VPNs), and secure websites (**HTTPS**). By encrypting the data being transmitted, encryption helps to ensure that the information remains confidential and secure, even when it is transmitted over public networks like the internet.

**Overall**, encryption plays a crucial role in ensuring the privacy and security of data transmitted over the internet, and is an essential tool for protecting sensitive information from unauthorized access.

## Explain the difference between a star and a mesh network

A **star network** and a **mesh network** are two different types of **network topologies**, or arrangements of clients and links in a network.

In a **star network**, all **clients** are connected to a central **switch**. The **switch** acts as a central point of communication, **forwarding data from one client to anothe**r. If one client fails, it only affects that client and the rest of the network continues to function normally. **Star networks** are simple to set up and manage, and are **commonly used in small networks**, such as in homes and small offices.

In a **mesh network**, on the other hand, **each client is connected to multiple other clients**, forming a network of interconnected clients. Data can be transmitted from one client to another by following **multiple paths**, rather than relying on a central client. This makes mesh networks **more resilient to failures**, (**more fault-tolerant)** as the network can continue to function even if one client fails. However, mesh networks can be more **complex** to set up and manage, and are typically used in larger, more complex networks.

**In summary,** the main difference between a star network and a mesh network is the arrangement of clients and links. Star networks have a central switch that acts as a hub for communication, while mesh networks have multiple interconnected clients, allowing for multiple paths for data transmission.

## Explain how anti malware software works to protect systems from cyber attacks

Anti-malware software works by **scanning the computer's files**, **memory**, and **network activity** to identify and neutralize any **potential threats**. The software uses a **database of known malware signatures** and behaviour patterns to detect and isolate malicious programs, such as **viruses, Trojans, and spyware**.

When a threat is detected, the **anti-malware software** will either remove the **malicious** software or quarantine it, so it cannot cause harm to the system. Some anti-malware programs will also provide **real-time protection** by constantly monitoring the system for signs of malicious activity. This can help prevent new infections from taking hold on the system.

Additionally, anti-malware software may also have **features** such as email filtering, web filtering, and firewall protection, which can help protect against new threats and attacks.

**In summary**, anti-malware software provides a first line of defence against cyber attacks by scanning for and removing malicious software and by implementing security measures to prevent future attacks.

## Describe the role of a signature file in anti-malware software

A **signature file** in anti-malware software is a database of known malware "signatures" that the software uses to identify and remove malicious software from a computer. The signature is a unique set of characteristics or "markers" that are specific to each type of malware.

When the anti-malware software scans a computer, it compares the files on the computer to the signatures in its database. If the software finds a match between a file on the computer and a signature in its database, it will flag the file as malicious and take action to remove it. This could include quarantining the file, deleting it, or repairing any damage it has caused.

The role of a signature file in anti-malware software is crucial in protecting a computer from malware. By using a database of known malware signatures, the software is able to quickly and effectively detect and remove malicious software, reducing the risk of a cyber attack. However, it is important to note that new types of malware are constantly being developed, and the anti-malware software must be **updated** regularly to ensure it has the latest signatures in its database.

## Explain the role of a firewall in preventing hacking (unauthorised access)

A **firewall** is a **security system** that monitors and **controls incoming and outgoing network traffic** based on a **set of security rules**. Its main purpose is **to prevent unauthorized access** to or from a private network.

A **firewall** acts as a **barrier** between a **trusted internal network and untrusted external networks**, such as the **Internet**. It inspects each incoming and outgoing **network packet** and only allows those that meet the **specified security criteria** to pass through. For example, a firewall may block incoming traffic from known malicious IP addresses, or block outgoing traffic to known malicious websites.

**In summary**, a firewall is an essential component of a network's security infrastructure and plays a crucial role in preventing hacking by controlling and monitoring network traffic and **blocking unauthorized access** to the network.

## Explain the key features of the Computer Misuse Act.

The UK Computer Misuse Act is a law enacted in 1990 to address **computer-related crime**. The act outlines the offenses that are illegal in relation to computers, **including hacking, unauthorized access to computer systems and data, and the distribution of malicious software.**

**The key features of the UK Computer Misuse Act include:**

**Unauthorized access to computer systems**: The act makes it illegal to gain unauthorized access to computer systems or data without permission. This includes hacking into someone else's computer or accessing computer systems without the owner's permission.

**Unauthorized modification of computer material**: The act makes it illegal to alter or delete computer data without authorization. This includes the modification of software or the deletion of important files.

**Unauthorized use of computer systems**: The act makes it illegal to use someone else's computer systems without permission, even if you do not cause any damage.

**Distribution of malicious software:** The act makes it illegal to create, distribute, or sell malicious software, such as **viruses** and Trojans, with the intention of causing harm to computer systems.

**Penalties**: Offenses under the UK Computer Misuse Act carry serious penalties, including fines and imprisonment. The maximum sentence for the most serious offenses is 14 years in prison.

In **summary**, the UK Computer Misuse Act is designed to protect computer systems and data from unauthorized access, modification, and use, and to punish those who engage in illegal computer-related activities.

## Explain the computational thinking principle of decomposition, with examples.

**Decomposition** is a principle of **computational thinking** that involves **breaking down complex problems** into smaller, more manageable parts. It involves dividing a large, complex problem into smaller, simpler components that can be solved independently and then recombined to solve the larger problem. An example of decomposition in programming is a **sub program**.

- A **sub program** is a small, self-contained piece of code that performs a **specific task**. In terms of decomposition, a sub program represents a way of breaking down a larger problem into smaller, more manageable parts.
- **For example**, consider a program that performs a complex calculation. This calculation can be decomposed into several smaller, independent steps. Each of these steps can be implemented as a separate sub program. The main program calls each sub program in turn to perform the calculation.
- By breaking down the problem into smaller parts, each sub program can be designed, tested, and debugged independently. This makes the development process more manageable and less prone to errors. Additionally, sub programs can be reused in other programs, reducing the amount of code that needs to be written and increasing code reuse.

## Explain the computational thinking principle of abstraction, with examples.

**Abstraction** is a principle of computational thinking that **involves ignoring irrelevant details** and **focusing only on the essential information**. It involves **simplifying** complex systems and problems by removing unnecessary details and only considering the most important aspects.

**For example**, when coding a dice roll the only important aspect is the generation of a random number between 1 and 6. All other physical aspects of the dice and the act of rolling the dice can be ignored.

Another common **example of abstraction** in **programming** is the use of **functions**. A function is a self-contained piece of code that performs a specific task and **returns** a result. When a function is called, the caller is only concerned with the inputs to the function and the result it returns, rather than the details of how the function performs its task. **For example,** the functions print() and len() in Python.

## Explain what social engineering is and how organisations and individuals can protect against it

**Social engineering** refers to the psychological **manipulation of people** into performing actions or divulging confidential information. It is often carried out through **false pretences** or impersonation and can take many forms, such as **phishing scams, baiting, pretexting, or quid pro quo**.

**To protect** against social engineering, organizations and individuals can follow these best practices:

**Awareness**: Educate employees and yourself about the various forms of social engineering and how to identify them.

**Verify information**: Always verify information before taking action or giving out sensitive information, especially if the request comes from an unexpected or unfamiliar source.

**Strong passwords**: Use strong passwords and change them regularly.

**Secure systems:** Ensure that all systems, including computers and mobile devices, are secured with up-to-date anti-virus software, firewalls, and encryption.

**Limit information sharing**: Be careful about the amount of personal and sensitive information you share online or with others.

**Use two-factor authentication**: Whenever possible, enable two-factor authentication to add an extra layer of security to your accounts.

**Be sceptical:** Be suspicious of unsolicited emails, phone calls, or messages, especially those that ask for personal information or pressure you to take immediate action.

Remember, social engineering attacks can be very convincing and manipulative, so it is important to be vigilant and always double-check before taking any action or sharing any information.

## Describe baiting, pretexting and quid pro quo cyber attacks.

"**Baiting**" in cyber security refers to a type of **social engineering** attack where an attacker leaves a **tempting item, such as a USB drive**, in a location where it is likely to be picked up by a target. The USB drive is loaded with malware that infects the target's computer when the drive is plugged in.

"**Pretexting**" is a type of **social engineering** attack where the attacker creates a **fake scenario** in order to **trick** the target into divulging confidential information. For example, an attacker may pose as a customer service representative from a bank and ask the target to provide their account number and password.

"**Quid pro quo**" is a type of **social engineering attack** where the attacker offers something of value to the target in exchange for something they want, such as access to confidential information. **For example**, an attacker may offer to fix a computer issue for a target in exchange for their login credentials.

In all three cases, the attacker uses **social engineering techniques** to trick the target into giving up confidential information or performing actions that compromise their security. These attacks are often successful because the attacker takes advantage of the trust that individuals have in others. To defend against these types of attacks, it is important to be **cautious** of unexpected or unsolicited requests for information, and to **verify** the identity of anyone asking for sensitive information.

## Explain the difference between open source and proprietary software and the advantages and disadvantages of each

**Open source and proprietary software** are two different types of software that are developed and distributed differently.

**Open source software** is software whose **source code is freely available** to the public, allowing anyone to modify or distribute the software. This means that the software is typically developed collaboratively by a community of developers, who can contribute to and improve the software**. Examples of popular open source software** include the Linux operating system, GIMP and LibreOffice.

**Proprietary software**, on the other hand, is software that is **owned by a company or individual**, who controls the distribution and modification of the software. The **source code of proprietary software is usually kept secret**, and users are only able to use the software as it is, without being able to make changes. **Examples of proprietary software** include Microsoft Windows, Apple macOS, and Adobe Photoshop.

**Advantages of open source software include:**

- **Lower cost**: Open source software is often free to use, and there are no licensing fees.
- **Flexibility**: Users can modify the software to fit their specific needs, and there are usually many different versions available.

**Advantages of proprietary software include:**

- **Reliability**: Proprietary software is often more reliable because it is developed and maintained by a single company, which can ensure that it works as intended.
- **Support**: Proprietary software usually comes with technical support and customer service, which can be helpful for users who need assistance.
- **Innovation**: Proprietary software companies often invest in research and development, which can lead to new and innovative features..

## Explain the disadvantages of open source and proprietary software

**Disadvantages of open source software include:**

- **Lack of support**: Open source software may not come with technical support or customer service, and users may have to rely on online communities for help.
- **Lack of consistency**: With open source software, there may be many different versions available, which can lead to compatibility issues and a lack of consistency.

**Disadvantages of proprietary software include**:

- **Cost**: Proprietary software can be expensive to purchase and use, especially for large organizations.
- **Lack of flexibility**: Users are not able to modify the software, which can limit its usefulness.
- **Dependence on a single entit**y: If the company that owns the proprietary software goes out of business or stops supporting the software, users may be left without a solution.

## Explain methods of preventing cyber attacks

**Cyber attacks** are a growing threat to individuals, businesses, and governments. There are several methods that can be used to prevent cyber attacks, including:

Use **strong passwords**: A strong password is a complex combination of letters, numbers, and symbols that is difficult to guess or crack. Users should use strong, unique passwords for all their online accounts, and change them regularly.

Keep **software up to date**: Regularly updating software and operating systems helps to fix known **security vulnerabilities** that could be exploited by attackers.

**Use antivirus software**: Antivirus software scans a computer's files and network traffic for known viruses and other malicious software, and can prevent these threats from infecting a system.

**Enable two-factor authentication**: Two-factor authentication adds an extra layer of security by requiring users to provide two forms of identification, such as a password and a fingerprint or a security token, to access their accounts.

**Back up important data**: Regularly backing up important data, such as documents and photos, helps to protect it in case of a cyber attack, hardware failure, or other problem.

**Be cautious when opening email attachments or clicking on links**: Email is one of the most common ways that attackers spread malware. Users should be cautious when opening attachments or clicking on links in emails, especially if the email is from an unknown sender.

**Educate yourself and others**: Staying informed about the latest cyber threats and best practices for preventing cyber attacks can help individuals and organizations better protect themselves.

By taking these preventive measures, individuals and organizations can reduce their risk of becoming the victims of a cyber attack, and help to keep their data and systems secure.

## Explain an Acceptable Use Policy and why an organisation would have one

An **Acceptable Use Policy (AUP)** is a set of rules that outline how employees and users of an organization's network, computers, and other resources should use them. The **purpose** of an AUP is to ensure that these resources are used for legitimate, productive, and ethical purposes, and to prevent activities that could harm the organization or its reputation.

**An AUP typically outlines**:

- What types of activities are allowed and what types of activities are prohibited.
- Who is responsible for the security of the organization's information and systems.
- How personal information should be protected and used.
- What the consequences are for violating the AUP.

Having an AUP is important for organizations because it **sets clear guidelines for employees and users**, **reducing the risk of security breaches, misuse of resources, or damage to the organization's reputation**. It also helps to establish a culture of **responsible use**, ensuring that everyone understands their role in protecting the organization's assets and information.

## Explain why employees should not use attach their own devices or portable storage to a network

Employees should not use their own devices or portable storage to a network for several reasons:

**Security risk**: Personal devices and portable storage may contain **malware**, viruses, or other security threats that could compromise the security of the network. Additionally, personal devices may not have the **same level of security as the devices provided by the organization, making them vulnerable to attack.**

**Data privacy**: Personal devices and portable storage may not have the **same level of data protection** as the organization's network, making it easier for sensitive or confidential information to be accessed, intercepted, or stolen.

**Lack of control:** When employees use their own devices or portable storage, the organization loses control over the data stored on these devices and the way they are used. This could lead to **unauthorized access or use of sensitive or confidential information**.

For these reasons, it is recommended that employees use only the devices and storage provided by the organization, and follow the organization's security policies and procedures to minimize the risk of a security breach or data loss.

## Explain the difference between incremental and full back up and why it is important to make regular backups

A **backup** refers to the **process of copying and saving data to another location** such as an external hard drive or the Cloud, so that it can be **restored in the event of data loss or corruption.**

In simple terms, the difference between **incremental and full backups** is the **amount of data being backed up and the frequency of the backups.**

A fu**ll backup** is a **complete copy** of all data that is being backed up. This type of backup captures all the data, including any new or changed files, and is typically done on a regular schedule, such as once a week or once a month.

An **incremental** backup, on the other hand, **only backs up the data that has changed since the last backup**. For example, if you have a full backup of your data from Monday and an incremental backup on Wednesday, the incremental backup will only contain the data that has changed between Monday and Wednesday. Incremental backups are typically done more frequently than full backups, such as every day or every few hours.

It's important to make **regular backups** of your data to protect against **data loss due to hardware failure, malware attacks, or other disasters**. Backups allow you **to restore your data** if it is lost or damaged, so it's important to have a reliable backup strategy in place. Regular backups, whether they are full or incremental, help ensure that your data is protected and can be easily restored in the event of an emergency.

## Explain how to sort data using the bubble sort algorithm

The bubble sort algorithm is a **simple sorting method used to sort data** in ascending or descending order. The algorithm works by **comparing pairs of adjacent elements in a list**, and **swapping their positions if they are not in the correct order.** This process is repeated until all elements are in the correct order.

**Here is a simple explanation of the bubble sort algorithm:**

1. Start by comparing the first two elements in the list. If the first element is larger than the second element, swap their positions.
2. Move to the next pair of elements and repeat step 1. Continue this process until all elements have been compared and possibly swapped.
3. Repeat the entire process for as many times as necessary until all elements are in the correct order and the number of swaps in a pass is zero.
4. The number of times the process is repeated depends on the size of the data. The larger the data, the more passes are required.

The bubble sort algorithm is **not the most efficient sorting method, especially for large data sets**. However**, it is simple and easy to understand and program.**

## Explain how to sort data using the merge sort algorithm

The **merge sort algorithm is a divide-and-conquer** method used to **sort data** in ascending or descending order. It works by **dividing the data into smaller and smaller parts until each part only contains one element, then merging the sorted parts back together to form a sorted whole.**

Here is a simple explanation of the merge sort algorithm:

1. Divide the data into two halves.
2. Recursively sort each half of the data by dividing it into two halves again, until each part only contains one element.
3. Merge the two sorted halves back together into one sorted whole.
4. Repeat the process for each half until all of the data is sorted.

The merge sort algorithm is considered to be one of the more **efficient** sorting methods, **especially for large data sets.**

## Explain how to search data using the linear search algorithm

The **linear search algorithm** is a **simple** method used to **search for a specific element in a list** of data. It works by **comparing each element in the list one by one until it finds the desired element or determines that it is not in the list.**

Here is a simple explanation of the linear search algorithm:

1. Start at the first element in the list.
2. Compare the first element to the desired element.
3. If the elements are the same, return the position of the element in the list.
4. If the elements are not the same, move on to the next element in the list and repeat step 2 and 3.
5. Continue this process until either the desired element is found or it is determined that the element is not in the list.

The linear search algorithm is a **simple and straightforward** method for searching data, but it can be **slow for large data sets** as it requires checking each element one by one. It is **easier to understand and program** than the binary search.

## Explain how to search data using the binary search algorithm

**The binary search algorithm** is a more **efficien**t method for searching data, but it **only works if the data is already sorted.** It works by **dividing the data into smaller and smaller parts and eliminating half of the remaining possibilities with each comparison.**

Here is a simple explanation of the binary search algorithm:

1. Find the middle item in the list.
2. Compare the middle item of the data to the search data.
3. If the middle item is the item being searched for, return its position in the list and stop the algorithm.
4. If the middle item is larger than the search data, repeat the search process in the left half of the data.
5. If the middle item is smaller than the search data, repeat the search process in the right half of the data.
6. Repeat this process of checking the middle item until either the search data is found or it is determined that the item is not in the list.

The binary search algorithm is a **faster method** for searching data compared to the linear search algorithm, **especially for large data sets, as it eliminates half of the remaining possibilities with each comparison**. However, it **can only be used when the data is already sorted and it is harder to code.**

## Compare merge and bubble sort algorithms in terms of efficiency

Both **the bubble sort** and **merge sort** algorithms are methods used to **sort data**, but they differ in terms of efficiency.

**Bubble sort is a simple and straightforward** sorting algorithm that works by repeatedly swapping adjacent elements if they are in the wrong order until the list is sorted. While bubble sort is easy to understand and implement, it is **not the most efficient sorting** algorithm. The **time it takes to sort the data increases rapidly as the size of the data increases.** This makes bubble sort a poor choice for large data sets. **Bubble sort is more efficient for memory use as it uses less memory than a merge sort.**

**Merge sort**, on the other hand, is a **more efficient sorting algorithm** that works by dividing the data into smaller parts and then merging those parts back together in a sorted order. **Merge sort is much faster** than bubble sort. **The time it takes to sort the data increases much more slowly as the size of the data increases**, making merge sort a good choice for large data sets. However **merge sort** is less **efficient in terms of memory** use as it uses more memory.

## Compare linear and binary search algorithms in terms of efficiency

Both **linear search and binary search** are algorithms used to search data, but they differ in terms of efficiency.

**Linear search** is a **simple search** algorithm that works by sequentially checking each element in a list until the desired element is found or it is determined that the element is not in the list. With a linear search the **time it takes to search the data increases linearly** with the size of the data. This makes linear search a **good choice for small data sets, but a poor choice for large data sets.**

**Binary search**, on the other hand, is a **more efficient search algorithm** that works by dividing the data into smaller and smaller parts and eliminating half of the remaining possibilities with each comparison. With a binary search the **time it takes to search the data increases much more slowly** as the size of the data increases. This makes binary search a good choice for large data sets, but it **can only be used when the data is already sorted**.

**In conclusion**, while linear search is a simple search algorithm that is easy to understand and implement, it is not very efficient for large data sets. Binary search, on the other hand, is a more efficient search algorithm that is well suited for large data sets, but it requires the data to be sorted.

## Describe the role of utility software, with examples

**Utility software** is a type of computer software that provides the user with **tools to perform specific tasks** on a computer.

**Examples of utility software include:**

**Antivirus software**: This type of software is designed to protect the user's computer from malware and other security threats.

**Backup software**: This type of software is used to make copies of important files and data in case they are lost or damaged.

**Disk cleanup software**: This type of software is used to remove unnecessary files and free up space on the user's computer.

**Disk defragmentation software**: This type of software is used to rearrange fragmented data on a magnetic hard drive to improve the computer's performance.

**System monitoring software**: This type of software is used to monitor the performance of the computer and identify potential problems.

**File compression software**: This type of software is used to reduce the size of large files so that they can be stored more efficiently and/or easily transferred from one computer to another.

**In conclusion,** utility software provides the user with a range of tools to perform specific tasks on their computer, helping them to manage, maintain, and optimize their computer more effectively.

## Explain the Internet of Things and the privacy concerns surrounding it

**The Internet of Things (IoT)** refers to the interconnected network of physical devices, vehicles, home appliances, and other items that are embedded with electronics, software, and sensors, allowing them to collect and exchange data. The IoT allows these **devices to communicate with each** other, share information, and respond to changing conditions and user inputs.

However, the widespread use of IoT devices **raises privacy concerns**, as **the collection and exchange of data can also be used to gather personal information about individuals**, such as their location, habits, and preferences. This information can be used for **targeted advertising**, but it can also be used for more **malicious** purposes, such as identity theft or other forms of **cybercrime**.

In order to address these privacy concerns, it is important for organizations to implement strong privacy policies and to secure the data collected by IoT devices. This can be done through encryption, secure data storage, and regular software updates, among other measures. Additionally, individuals can take steps to protect their privacy by carefully researching the privacy policies of the devices they use, being cautious about the information they share online, and using security software to protect their devices and data.

In conclusion, the IoT offers many benefits, but it also raises significant privacy concerns. Organizations and individuals must work together to ensure that the data collected by IoT devices is used responsibly and securely.

## Give some examples of devices connected to the Internet of Things

The Internet of Things (IoT) is a **network of connected devices** that **collect and exchange data**. Here are some examples of IoT devices:

**Smart home devices**, such as smart thermostats, smart locks, smart lights, and smart plugs

**Wearables**, such as fitness trackers, smartwatches, and health monitors

**Smart appliances**, such as refrigerators, washing machines, and ovens

**Security devices**, such as security cameras, motion sensors, and smoke detectors

**Transportation devices**, such as GPS tracking devices for cars, smart bike locks, and wearable devices for snow sports

**Agricultural devices**, such as soil moisture sensors and weather monitoring systems

**Environmental monitoring devices**, such as air quality sensors and water quality monitoring systems

## Compare different storage media used for secondary storage

Computers use different media to store data, and the most common methods are magnetic, optical, and solid-state storage.

**Magnetic storage** uses magnetic material to store data. The most common type of magnetic storage is the **hard disk drive (HDD**), which uses spinning disks **coated with magnetic material** to store data. Data is stored on the disk in the form of **magnetic patterns to represent the binary 1s and 0s**. HDDs **are relatively cheap and have large storage capacities**, but they are also **slow** and can be affected by **physical damage or wear and tear**.

**Optical storage** uses **light** to store data. The most common type of optical storage is the compact disc (CD), which uses a laser to read and write data to a disc. When data is stored on a CD, it is encoded in a series of tiny **pits and lands** on the surface of the disc. These represent the **1s and 0s.** Pits and lands **reflect light differently** allowing the data to be read back. CDs are **relatively cheap and have good durability**, but they have **limited storage capacities and slow read/write speeds**.

**Solid-state storage**) stores data using electronic memory chips. Binary data is stored using **electrical charges**. Unlike traditional mechanical hard drives, SSDs have **no moving parts**, which makes them **faster, more reliable, and more durable**.. The most common type of solid-state storage is the **solid-state drive (SSD), which uses NAND flash memory to store data**. SSDs are **faster** and **more reliable than HDDs**, but they are also **more expensive.**

## Describe the ‘stored-program’ concept and the Von Neumann architecture

The **"stored-program**" concept refers to the idea that a **programs data and instructions** can be stored in the computers **primary memory (RAM). This means** that a computer can change the program it is running simply by changing the contents of its memory, allowing it to perform **a wide range of tasks** without having to be physically reconfigured.

The **Von Neumann architecture** is a computer architecture design **that implements the stored-program concept** and is named after mathematician and computer scientist John von Neumann. The Von Neumann architecture **consists of four main components**: the central processing unit (**CPU), the memory, the input/output (I/O) devices, and the buses** that connect these components.

In the Von Neumann architecture, the **CPU reads the instructions stored in memory** and executes them, while **the memory stores both the instructions and the data** the program operates on. The **I/O devices allow the computer to receive inputs and produce outputs**, while **the buses connects the different components and enables communication between them.**

The Von Neumann architecture has been the basis for most computers since the 1950s and remains the most widely used computer architecture design today. Its simplicity and versatility have made it an enduring and influential design in the history of computing.

## Describe the Fetch-Decode-Execute cycle with reference to the buses

The **fetch-decode-execute cycle** is the basic process that a computer follows to execute a program. The process involves communication between the central processing unit (CPU), memory, and input/output (I/O) devices through three main types of buses: the control bus, address bus, and data bus.

**Here's how the fetch-decode-execute cycle works**:

**Fetch**: The **CPU sends an address to memory via the address bus**, which specifies the location of the next instruction to be executed. **The memory retrieves the instruction and sends it to the CPU via the data bus.** The instruction is stored in a **register** in the CPU.

**Decode**: The **Control Unit of the CPU decodes** the instruction to determine what operation it represents and what data is required to perform the operation.

**Execute**: The ALU performs the operation specified by the instruction, using the data retrieved in the fetch and decode stages. The ALU is responsible for performing arithmetic and logical operations as specified by the instruction decoded by the control unit.

The **control bus is used to send commands to memory or I/O devices, specifying what data should be read or written**. The **address bus** is used to specify the location of the data in memory, and the **data bus** is used to send the data to or from the CPU. The result of the operation is stored in memory.

## Explain the different levels of user access rights with examples of each

**User access rights** refer to the **level of permission** that a user has to perform certain actions within a system. There are **typically three levels of user access rights**:

**Administrative rights**: These are the **highest** level of access rights, granting the user **complete control** over the system. **Examples of actions** that a user with administrative rights can perform include installing software, configuring system settings, and creating and deleting user accounts.

**User rights:** These are **more limited** than administrative rights, but still allow the user to perform certain actions. **Examples of actions** that a user with user rights can perform include creating and editing files, printing documents, and accessing the internet.

**Guest or limited rights**: These are the **lowest level** of access rights, granting the user the least amount of control over the system. **Examples of actions** that a user with guest or limited rights can perform include only being able to view files, not being able to make changes to the system, and being restricted in their internet access.

**Organizations use different levels of user access right**s to control who can perform certain actions and to protect sensitive information. For example, a financial institution might grant administrative rights only to a few trusted employees, while granting user rights to most employees, and guest or limited rights to visitors.

## Explain the differences between a Local Area Network (LAN) and a Wide Area Network (WAN)

A **computer network** is a collection of interconnected devices, such as computers, servers, and other hardware, that are **able to communicate with each other and exchange data**. The purpose of a computer network is to **allow the sharing of resources**, such as printers, files, and internet access, among multiple devices, as well as making communication between users easier.

There are two main types of computer networks: Local Area Networks (LANs) and Wide Area Networks (WANs).

**A Local Area Network (LAN)** is a network that is confined to a **small geographical area**, such as a home, office, or building. The organisation is **responsible for its own hardware and cabling.**

**A Wide Area Network (WAN)** is a network that covers a **large geographical area**, such as a city, country, or even the entire world. WANs are used for large-scale operations, such as connecting multiple offices in different locations or connecting to the internet. **WANs use the hardware owned by a third-party** such as the cables and routers managed by an internet service provider (ISP)

**In summary,** a computer network is a collection of interconnected devices that are able to communicate with each other and exchange data. Local Area Networks (LANs) are confined to a small geographical area and are used for small-scale operations, while Wide Area Networks (WANs) cover a large geographical area and are used for large-scale operations.

## Explain the difference between Wi-Fi and Bluetooth

**Wi-Fi** and **Bluetooth** are two different **wireless** technologies that are used for different purposes.

**Wi-Fi** is a technology that **allows devices to connect to a wireless network** and access the internet. It is used to connect laptops, smartphones, tablets, and other devices to the internet via a **Wireless Access Point**. Wi-Fi is typically **faster than Bluetooth** and has a **longer range**, making it ideal for use in homes, offices, and public places where a large number of devices need to connect to the internet.

**Bluetooth**, on the other hand, is a **short-range** wireless technology that is used to **connect devices to each other**, allowing them to share data and communicate. Bluetooth is typically used for connecting devices such as headphones, speakers, and smartwatches to smartphones and other devices. Bluetooth **is slower than Wi-Fi and has a shorter range**, but it is **more power-efficient** and **does not require a separate network connection.**

In **summary**, Wi-Fi is used for connecting devices to the internet, while Bluetooth is used for connecting devices to each other.

## Explain how a ZigBee mesh network works

**ZigBee** is a **low-power, low-data-rate wireless** technology that is used to **connect devices** in a smart home or **Internet of Things (IoT)** network. It is designed for **short-range communication** and is used for devices that require low data rates, such as sensors, light switches, and thermostats.

**ZigBee** operates on **a mesh network**, which means that data is passed from device to device until it reaches its destination. **This allows devices to communicate even if they are not in direct range of a central hub or router, making it a reliable and scalable solution for IoT networks.**

One of the key benefits of ZigBee is its **low power consumption**, which allows devices to run on batteries for several years. It is also **secure, easy to set up, and compatible with a wide range of devices**, making it a popular choice for smart home and IoT applications.