**Area of study: 1.1 Programs and applications**

**Lessons**

| **Lesson number** | **Specification coverage** | **GLH** | **Lesson aims and outcomes** | **Lesson ideas, key words and activities** | **Useful resources** | **Student independent learning – ideas and useful resources** |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | 1.1  Programs | **1** | By the end of the lesson students should:   * know what a ‘program’ is * know the characteristics of a program * understand the purpose of a program. | **Starter**  Think pair share: ask students to think about different types of software they use daily (e.g., games, word processors, web browsers) and write them on the board.  **Main task**  Teach students what a **program** is, the characteristics and the purposes of certain programs. Explain to students that a program does not necessarily have an end user or a purpose commercially, but maybe a script or file a programmer may construct.  Give students a range of different scenarios, get them to categorise if they are a **program** or not.  Group Work: divide students into small groups and assign each group a type of program (e.g., utility, game, educational software).  Each group should identify the characteristics and purpose of their assigned program type and present their findings to the class.  **Alternative task**  If possible, create a batch file or script to automate a task such as creating folders for students. Illustrate that this isn’t something that one would purchase, but technical users can do to improve their workflow.  Design, program and test an automation file. | [Application vs. Program (What's the Difference?) - Rapid Blog](https://rapidapi.com/blog/application-vs-program/)  [Is There any Difference between a “Program” and an “Application“?](https://flatrocktech.com/blog/is-there-any-difference-between-a-program-and-an-application)  [How to Create and Run a Batch File in Windows 10 and 11](https://www.makeuseof.com/tag/write-simple-batch-bat-file/#:~:text=To%20create%20a%20Windows%20batch%20file%2C%20follow%20these,the%20BAT%20file%20you%20just%20created.%20More%20items) | Students should use Cornell note taking to create structured notes to reflect on the lesson.  Create a glossary of definitions to add to, using flash cards or paper. |
| **2** | 1.1  Applications | **1** | By the end of the lesson students should:   * know what an application is * know the characteristics of an application * know the function of an application * understand the relationship between programs and applications. | **Starter**  Begin with a brief discussion to introduce the concept of **applications**. Ask students to share examples of **applications** they use daily and what they use them for.  **Main task**  Get the students to create a comparison table, reflecting on both this and the previous lesson.  Students should compare the characteristics of **applications** and **programs** (see key concepts table).  Ask students to choose one **application** from their comparison table and analyse its features and purpose in more detail.  They can create a short presentation or a poster to share their findings with the class.  Create a quiz using an interactive platform like [Kahoot!](https://kahoot.com/) or [Quizizz](https://quizizz.com/).  Include questions that help students define what an application is, identify its **characteristics**, and understand its purpose.  **Alternative task**  Ask students to generate ideas for a new application that could solve a problem or fulfil a need they have identified.  They can create a basic outline or storyboard of their **application**, detailing its features and purpose | [Application vs. Program (What's the Difference?) - Rapid Blog](https://rapidapi.com/blog/application-vs-program/)  [Is There any Difference between a “Program” and an “Application“?](https://flatrocktech.com/blog/is-there-any-difference-between-a-program-and-an-application) | Students should use Cornell note taking to create structured notes to reflect on the lesson.  Create a glossary of definitions to add to, using flash cards or paper. |
| **3** | 1.1  Types of devices using programs/applications | **1** | By the end of the lesson students should:   * list devices that programs and applications use * know the characteristics of an application * understand that some programs and applications are suited to different devices better than others. | **Starter**  Begin by discussing the different types of **digital devices** that students might have at home. Highlight the variety of **devices** and their uses, such as smartphones, tablets, laptops, smart TVs, and gaming consoles.  **Main task**  Get students to create a mind map of all the **digital devices** they have at home.  Expand the mind map to list the various **programs** or **applications** that run on those **devices**.  Have students create a comparison table to reflect on both this and the previous lesson.  Students should compare **the characteristics of applications** and **programs**. Use the key concepts table to guide this comparison. For instance, they can compare aspects like functionality, user interface, and purpose.  **Alternative task**  Ask students to choose one **device** from their mind map and analyse how often they use it and for what purposes.  They can create a pie chart or bar graph to represent their usage patterns. | [Teach ICT - GCSE ICT - types of computer](https://www.teach-ict.com/gcse_new/computer%20systems/types_computer/miniweb/index.htm) | Students should consolidate their learning by creating flashcards on devices and their uses. |

**Subject knowledge support for this area of study**

| **Subject knowledge enhancement** | **Details** |
| --- | --- |
| **Underlying knowledge and understanding** | Students should have a basic understanding of the concept of applications and algorithms, as well as types of devices used. |
| **Common misconceptions** | Programs can be created for different reasons: to solve problems, automate tasks, or for entertainment.  Highlight that not all programs are designed for end-users; some are tools for developers (e.g., scripts, libraries). |
| **Key concepts** | **Program:** a program is a set of instructions that a computer follows to perform a specific task. It can be as simple as a script that adds two numbers or as complex as an operating system. Programs are the building blocks of software and can run independently or as part of a larger system.  **Application:** an application (or app) is a type of program designed for end-users to perform specific tasks. Applications are typically more user-friendly and are built to help users complete activities like word processing, browsing the internet, or playing games. Essentially, all applications are programs, but not all programs are applications.  **Characteristics of a program:**   * set of instructions: a program consists of a sequence of instructions that a computer follows to perform a specific task * execution: programs are executed by the computer's CPU * purpose: programs can be created for various purposes, such as solving problems, automating tasks, or performing calculations * independence: programs can run independently or as part of a larger system * flexibility: programs can be simple (like a script) or complex (like an operating system).   **Characteristics of an application:**   * user-focused: applications are designed with end-users in mind, providing a user-friendly interface * specific tasks: applications are created to help users perform specific tasks, such as word processing, browsing the internet, or playing games * integration: applications often integrate multiple programs to provide a cohesive user experience * installation: applications usually come with an easy installation process and can be easily uninstalled * commercial purpose: many applications are developed for commercial use, aiming to meet the needs of consumers or businesses.   **Devices that use programs and applications**  **Computers**  Desktops and laptops: these are used for a wide range of tasks, from professional work to gaming and personal use. They run operating systems like Windows, macOS, and Linux.  Servers: these are powerful computers that provide services to other computers over a network. They run specialized server operating systems**.**  **Mobile devices**  Smartphones: these are handheld devices that combine a phone with a computer. They run mobile operating systems like Android and iOS.  Tablets: larger than smartphones, tablets are used for browsing, media consumption, and productivity tasks.  **Wearables**  Smartwatches: these are wrist-worn devices that can run apps, track fitness, and provide notifications.  Fitness trackers: these devices monitor physical activity and health metrics.  **Home devices**  Smart home devices: these include smart speakers, thermostats, and security systems that can be controlled via apps.  Smart TVs: these televisions can run streaming apps and connect to the internet.  **Gaming consoles**  Consoles: devices like the PlayStation, Xbox, and Nintendo Switch run games and entertainment apps.  **Industrial and specialized devices:**  Medical devices: equipment used in healthcare, such as MRI machines and patient monitors, run specialized software.  Automotive systems: modern cars use software for navigation, entertainment, and vehicle diagnostics.  **Embedded systems**  Appliances: many household appliances, like microwaves and washing machines, use embedded software to operate.  IoT devices: internet of things devices, such as smart sensors and connected appliances, run specialized software to communicate and perform tasks. |
| **Subject knowledge enhancement for teachers** | Read: [Application vs. Program (What's the Difference?) - Rapid Blog](https://rapidapi.com/blog/application-vs-program/)  Watch: [What is Software](https://www.youtube.com/watch?v=PBDlbSAddAI)  Listen: [Software Engineering Daily | Podcast on Spotify](https://open.spotify.com/show/6UCtBYL29hwhw4YbTdX83N)  Refer to the OCR endorsed textbook. |