TOUCHPAD

Information Technology (V.5.0)

8

Teacher's Manual

Extended Support for Teachers

ORANGE
www.orangeeducation.in

Teacher's Time Table

Sessions / Days	0	I	II	III	IV	BREAK	V	VI	VII	VIII
Monday										
Tuesday										
Wednesday										
Thursday										
Friday										
Saturday										

Development Milestones in a Child

Development milestones are a set of functional skills or age-specific tasks that most children can do at a certain age. These milestones help the teacher identify and understand how children differ in different age groups.

Age
5 - 8 Years

Physical	• First permanent tooth erupts • Shows mature throwing and catching patterns • Writing is now smaller and more readable • Drawings are now more detailed, organised and have a sense of depth
Cognitive	• Attention continues to improve, becomes more selective and adaptable • Recall, scripted memory, and auto-biographical memory improves • Counts on and counts down, engaging in simple addition and subtraction • Thoughts are now more logical
Language	• Vocabulary reaches about 10,000 words • Vocabulary increases rapidly throughout middle childhood
Emotional/ Social	• Ability to predict and interpret emotional reactions of others enhances • Relies more on language to express empathy • Self-conscious emotions of pride and guilt are governed by personal responsibility • Attends to facial and situational cues in interpreting another's feelings • Peer interaction is now more prosocial, and physical aggression declines

"If you cannot do great things, do small things in a great way."

Age 9 - 11 Years

Physical	• Motor skills develop resulting in enhanced reflexes
Cognitive	• Applies several memory strategies at once • Cognitive self-regulation is now improved
Language	• Ability to use complex grammatical constructions enhances • Conversational strategies are now more refined
Emotional/ Social	• Self-esteem tends to rise • Peer groups emerge

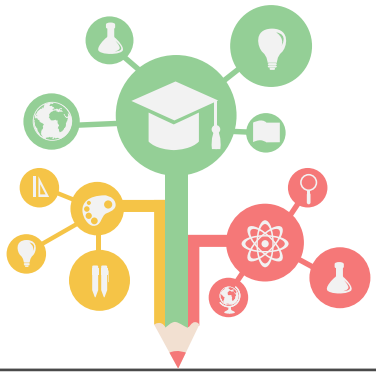
Age 11 - 20 Years

Physical	• If a girl, reaches peak of growth spurt • If a girl, motor performance gradually increases and then levels off • If a boy, reaches peak and then completes growth spurt • If a boy, motor performance increases dramatically
Cognitive	• Is now more self-conscious and self-focused • Becomes a better everyday planner and decision maker
Emotional/ Social	• May show increased gender stereotyping of attitudes and behaviour • May have a conventional moral orientation

Managing the children's learning needs according to their developmental milestones is the key to a successful teaching-learning transaction in the classroom.

"Family is the most important thing in the world."

Teaching Pedagogies



Pedagogy is often described as the approach to teaching. It is the study of teaching methods including the aims of education and the ways in which such goals can be achieved.

Lesson Plans

A lesson plan is the instructor's road map which specifies what students need to learn and how it can be done effectively during the class time. A lesson plan helps teachers in the classroom by providing a detailed outline to follow in each class.

A lesson plan addresses and integrates three key components:

- Learning objectives
- Learning activities
- Assessment to check the student's understanding

A lesson plan provides an outline of the teaching goals:

Before the class

1. Identify the learning objectives.
2. Plan the lesson in an engaging and meaningful manner.
3. Plan to assess student's understanding.
4. Plan for a lesson closure.

During the class

Present the lesson plan.

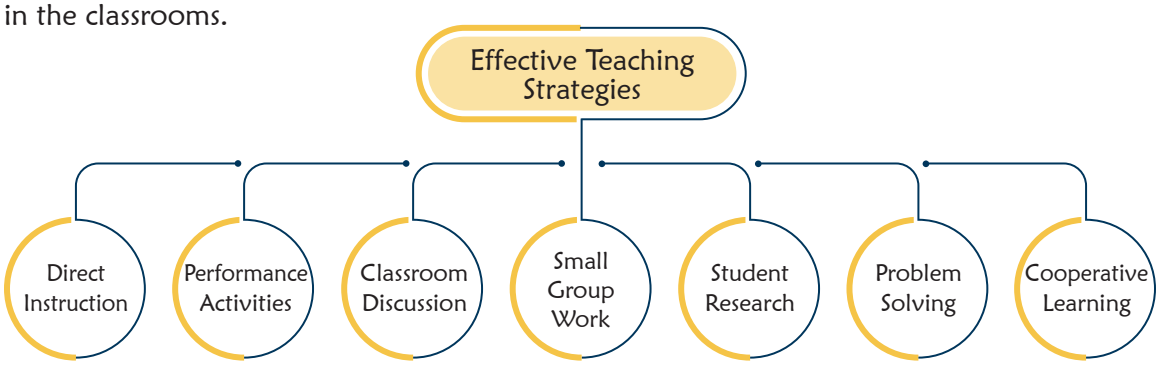
After the class

Reflect on what worked well and why. If needed, revise the lesson plan.

"Knowing yourself is the beginning of all wisdom."

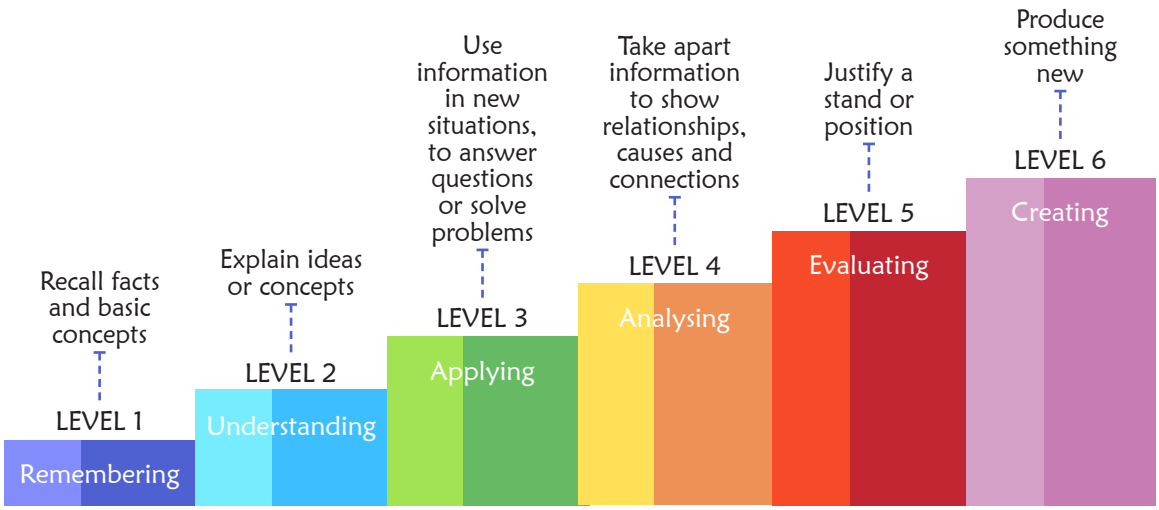
Teaching Strategies

Numerous strategies have evolved over the years to facilitate the teaching-learning process in the classrooms.



Bloom's Taxonomy

Bloom's Taxonomy was created by Dr Benjamin Bloom and several of his colleagues, to promote higher forms of thinking in education instead of rote learning. There are three domains of learning: cognitive (mental), affective (emotional), and psychomotor (physical). However, when we refer to Bloom's Taxonomy we speak of the cognitive domain. Bloom's Taxonomy is a list of cognitive skills that is used by teachers to determine the level of thinking their students have achieved. As a teacher, one should attempt to move students up the taxonomy as they progress in their knowledge.



Teachers should focus on helping students to remember information before expecting them to understand it, helping them understand it before expecting them to apply it to a new situation, and so on.

> "If you have no confidence in self, you are twice defeated in the race of life."

CLASS 8

LESSON PLAN

1 Fundamentals of Computer

Teaching Objectives

By the end of this lesson, students will be able to:

- Understand the fundamental principles and operations of a computer system, including how data is processed, stored, and retrieved.
- Differentiate between hardware and software and understand their roles in computing.
- Identify, compare, and explain the various types of input/output devices used in modern computer systems.
- Understand the functionality of disk management and file organization utilities, such as File Explorer, Disk Cleanup, and Recycle Bin.
- Develop an understanding of computer startup, shutdown, and maintenance processes, and recognize the importance of properly maintaining system performance.

Number of Sessions	
Theory	Practical
3	2

Teaching Plan

Engagement (Introduction)

1. **Question Prompt:** "What do you think is the most important part of a computer and why?"
2. **Interactive Activity:** Show a diagram of a computer and ask students to identify different components (CPU, RAM, hard drive, etc.) and their function.
3. **QAXP Activity:** In groups, students list different computer tasks (e.g., writing a report, designing a game) and map the hardware/software required for each.

Lesson Delivery

1. **What is a Computer?**
   - A computer is a programmable device that receives input, processes data, and produces output.
   - **Data Flow:** Introduce the concept of the IPO (Input-Process-Output) cycle in computing.

- **Example:** Show a video or demo that illustrates how a simple task like typing on a keyboard is processed by the CPU and displayed on the screen.

2. **Basic Components of a Computer System**
   - **Hardware:** Discuss different types of hardware: Input devices, Processing units, Output devices, and Storage devices.
   - **Software:** Introduce the role of system software (e.g., operating systems, drivers) and application software (e.g., word processors, games).
   - **Example:** Demonstrate the interaction between hardware and software using a simple program (e.g., opening a file and displaying its contents).
3. **Understanding Input Devices**
   - **Keyboard:** The most common text input device.
   - **Mouse:** Allows users to interact with graphical elements.
   - **Scanner and Microphone:** Convert physical data into digital form.
   - **Advanced Input Devices:** Discuss the use of touchpads, graphics tablets, and biometrics (e.g., fingerprint scanners).
   - **Activity:** Students will identify and test various input devices like a keyboard, mouse, and scanner in the computer lab.
4. **Understanding Output Devices**
   - **Monitor:** Visual output for text, images, and video.
   - **Printer and Plotter:** Output devices for producing hard copies of documents and graphics.
   - **Speakers/Headphones:** Output devices for audio.
   - **Activity:** Show students how to configure and test different output devices.
5. **Disk Management and File Explorer**
   - **Disk Management:** Explain the role of disk management in organizing storage space and optimizing disk performance.
   - **File Explorer:** Show how to navigate File Explorer, create folders, and manage files.
   - **Activity:** Students will practice creating, renaming, and organizing files into folders. They will also perform disk cleanup to remove unnecessary files and improve system performance.
6. **The Recycle Bin and Data Recovery**
   - **Recycle Bin:** Explain how files are temporarily stored in the Recycle Bin and how to restore them.
   - **File Restoration:** Show how to retrieve accidentally deleted files.
   - **Activity:** In groups, students will delete and restore files from the Recycle Bin.

7. **Proper Computer Shutdown and Maintenance**
    - **Shutdown Process:** Demonstrate the proper method of shutting down a computer, including saving work and closing programs.
    - **System Maintenance:** Teach students how to perform basic maintenance tasks like restarting the computer, cleaning the system cache, and updating software.
    - **Activity:** Students will practice shutting down and restarting computers correctly. They will also review their disk cleanup procedures.

Practical Activities

- **File Management Practice:** Students will organize their files into folders, practice moving, copying, and deleting files, and use search functions to locate specific files.
- **Disk Cleanup:** In pairs, students will run the Disk Cleanup utility on their computers and analyze the results.
- **Group Project:** Students will create a guide for maintaining a computer's performance, including best practices for file management, system updates, and disk management.

Extension

- **Creative Task:** Have students create a flowchart or infographic explaining the steps involved in organizing and maintaining computer files.
- **Discussion Prompt:** Ask students how well-managed file systems can improve productivity and efficiency in schoolwork or personal tasks.
- **Oral Questions:**
    - How does the interaction between input devices and software help perform tasks on a computer?
    - What steps can you take to protect your computer from storage overload?
    - Why is it important to empty the Recycle Bin regularly?
    - How do input devices influence the way we use computers for different tasks?
    - What impact does proper file organization have on your productivity?
    - How can poor system maintenance affect the overall performance of a computer?
    - What happens if you don't properly close programs before shutting down the computer?
    - Can a computer function without an operating system? Why or why not?
    - How does your computer's hardware affect its performance in gaming or graphic-heavy tasks?
    - Why is regular disk maintenance important for preventing data corruption?

Assessment and Evaluation

- **Quiz:** Multiple-choice and short-answer questions on hardware, software, disk management, and file systems.
- **Practical Test:** Have students demonstrate basic file management skills, including creating, renaming, and organizing files, and performing disk cleanup.
- **Peer Review:** Students will share their projects and discuss how they organized their files and folders.

Suggested Activity

- **Group Activity:** Create a class-wide guide for file organization and computer maintenance. Students will work in groups to present their guide using visuals and demonstrations.
- **Digital Drill:** Students will use File Explorer to complete a scavenger hunt by finding specific files based on certain search criteria (e.g., file type, name, or date modified).

2 Algorithm and Flowchar

Teaching Objectives

By the end of this lesson, students will be able to:

- Understand the problem-solving steps, including identifying and analyzing problems.
- Write algorithms following basic guidelines.
- Understand and implement control structures like sequential, selection, and repetition in algorithms.
- Create flowcharts to visually represent algorithms using standard symbols.
- Recognize the difference between algorithms and flowcharts and apply them in real-world problem-solving.

Number of Sessions	
Theory	Practical
3	2

Teaching Plan

Engagement (Introduction)

1. **Question Prompt:** "Have you ever had to solve a problem step-by-step? How did you approach it?"
2. **Interactive Activity:** Show a video or diagram of a simple task (like making tea) and ask students to list the steps involved in the process.
3. **QAXP Activity:** In groups, students discuss a problem they've recently faced and how they solved it, highlighting the steps involved.

Lesson Delivery

1. **Problem-Solving Steps**
   - Define problem-solving as breaking down a challenge into smaller steps to find a solution.
   - **Steps for Problem-Solving:**
     - **Identify the Problem:** Understand the issue by gathering information.
     - **Analyze the Problem:** Break it down into smaller parts and gather relevant data.
     - **Create Possible Solutions:** Brainstorm different ways to solve the problem.
     - **Evaluate and Choose the Best Solution:** Assess each solution's feasibility and effectiveness.
     - **Plan the Implementation:** Organize tasks and allocate resources.
     - **Implement the Solution:** Put the plan into action.
     - **Evaluate the Outcome:** Assess the success of the solution and refine it if needed.
2. **Writing Algorithms**
   - Define an algorithm as a step-by-step procedure for solving a specific problem.
   - **Guidelines for Writing Algorithms:**
     - Begin with "Start" and end with "Stop."
     - Use simple, understandable language.
     - It should follow a logical sequence.
     - The algorithm must terminate after a finite number of steps.
   - **Example Algorithm:**
     - Write an algorithm for making lemonade (refer to examples in Chapter A3).
3. **Control Structures in an Algorithm**
   - **Sequential Control Structure:** Steps are executed one after another in order.
   - **Selection Control Structure:** Conditional branching (e.g., if-else statements).
   - **Repetition Control Structure:** Repeating tasks (loops).
   - **Example Algorithms:**
     - Add two numbers.
     - Check if a number is positive or negative.
     - Calculate the area of a circle.
     - Display even or odd numbers from 1 to 20.
4. **Flowcharts**
   - Define flowcharts as visual representations of algorithms.

- **Symbols in Flowcharts:**
  - **Oval:** Start or Stop.
  - **Parallelogram:** Input or Output.
  - **Rectangle:** Process or Operation.
  - **Diamond:** Decision (Yes/No).
  - **Arrow:** Direction of flow.
- **Creating Flowcharts:** Guide students in drawing flowcharts using standard symbols.
  - **Example Flowchart:** Display the process for adding two numbers using Google Drawings.

5. **Difference Between Algorithm and Flowchart**
   - **Algorithm:** A step-by-step written set of instructions.
   - **Flowchart:** A graphical representation of the algorithm.
   - **Advantages of Flowcharts:**
     - Easier to understand and visualize.
     - Ideal for representing branching and looping.
   - **Advantages of Algorithms:**
     - Detailed and easy to convert into code.
     - Easier to test and find errors.

Practical Activities

- **Activity 1:** Write an algorithm for a simple task (e.g., making a sandwich, going to school).
- **Activity 2:** Create a flowchart based on the algorithm from Activity 1.
- **Activity 3:** Solve a real-world problem by writing an algorithm and drawing a flowchart for it (e.g., calculate the average of three numbers).
- **Group Project:** Create and present an algorithm and flowchart for a problem faced in school or at home (e.g., organizing files on a computer).

Extension

- **Creative Task:** Have students design an algorithm and flowchart for a more complex problem (e.g., organizing a birthday party or planning a trip).
- **Discussion Prompt:** Discuss when to use an algorithm and when to use a flowchart in solving problems. Why is visualizing a solution sometimes better than writing it down?
- **Oral Questions:**
  - What makes an algorithm different from a flowchart?
  - How does breaking down a problem into smaller steps help in solving it?

- Why is it important to test your algorithm before implementing it?
- Can an algorithm have more than one solution? Give an example.
- How can you ensure that your flowchart is easy to follow?

Assessment and Evaluation

- **Quiz:** Multiple-choice and short-answer questions on algorithms, flowcharts, and control structures.
- **Practical Test:** Students create an algorithm and flowchart for a given task, explaining each step.
- **Peer Review:** Students review and provide feedback on their peers' algorithms and flowcharts.

Suggested Activity

- **Group Activity:** Have students create flowcharts for different scenarios (e.g., finding the smallest number from a list, calculating the area of a rectangle).
- **Flowchart Challenge:** In groups, students create a flowchart to solve a problem, such as displaying all multiples of 3 between 1 and 30.

3 Learning Scratch

Teaching Objectives

By the end of this lesson, students will be able to:

- Understand the Scratch programming environment and its components.
- Create interactive programs using Scratch blocks for controlling sprites and adding animations.
- Design simple projects involving sprites, backdrops, and user inputs.
- Implement basic game mechanics and animations using Scratch.
- Explore event handling and broadcast messages for interactivity.

Number of Sessions	
Theory	Practical
3	2

Teaching Plan

Engagement (Introduction)

1. **Question Prompt:** "Have you ever tried creating a game or animation on your own? What steps would you follow to make it?"
2. **Interactive Activity:** Demonstrate a simple Scratch project (e.g., a sprite moving across the screen) and ask students how they think it works.
3. **QAXP Activity:** In small groups, students brainstorm ideas for simple interactive games or animations they could create with Scratch.

Lesson Delivery

1. **Introduction to Scratch**
   - **What is Scratch?** Explain that Scratch is a visual programming language designed for beginners, primarily used to create animations, stories, and games.
   - **Why Scratch?** Discuss the importance of learning programming through visual representation, where students can see how their code runs in real-time.
2. **Downloading and Installing Scratch**
   - Show the process of downloading and installing Scratch on various platforms (Windows, macOS, etc.).
   - **Activity:** Have students follow the instructions and set up Scratch on their computers or devices.
3. **Scratch Interface**
   - **Stage:** The area where the sprites perform actions.
   - **Sprites:** Visual objects that can be programmed to perform various tasks (moving, speaking, interacting with other sprites).
   - **Backdrops:** Set the scene for the Scratch projects.
   - **Coding Area:** Where students drag and drop blocks to create their programs.
   - **Blocks Palette:** Categorized blocks that perform specific functions (e.g., motion, control, events, operators).
4. **Sprite Manipulation**
   - Show how to add, resize, and delete sprites.
   - Activity: Have students add a sprite, resize it, and change its costume or color.
5. **Using Scratch Blocks**
   - **Motion Blocks:** Control the movement of sprites.
   - **Looks Blocks:** Change the appearance of sprites (e.g., say something, change costume).
   - **Sound Blocks:** Add sound effects or music to the project.
   - **Events Blocks:** Trigger actions based on user input (e.g., when the flag is clicked, when a key is pressed).
   - **Control Blocks:** Include loops and conditional statements for repeated actions or decisions in programs.
   - **Operators:** Perform mathematical or logical operations.
   - **Activity:** Guide students through creating a simple animation where a sprite moves when clicked.

6. **Event Handling**
    - **Broadcasting Messages:** Introduce how broadcast messages can trigger specific actions in Scratch.
    - **Example:** Use a broadcast message to switch between scenes in an animation or to trigger a sound when a sprite touches another sprite.
    - **Activity:** Have students create an event where one sprite broadcasts a message to make another sprite react (e.g., changing the sprite's costume or playing a sound).
7. **Creating Interactive Projects**
    - **Game Design:** Guide students in creating simple interactive games, like a sprite that moves when the arrow keys are pressed.
    - **Animation Design:** Show how to create an animation with multiple sprites and backdrops, triggered by user input (e.g., pressing the spacebar).
    - **Activity:** Students create a mini-game where a sprite needs to catch falling objects (use the "if" and "touch" blocks for interaction).

Practical Activities

- **Activity 1:** Students will create a basic interactive animation, using motion and looks blocks to animate a character moving across the screen.
- **Activity 2:** Students will design a simple game where a sprite moves using keyboard inputs and interacts with other objects on the stage.
- **Group Project:** In groups, students will design a more complex game or animation using Scratch's broadcast and event-handling features. Each group presents their project to the class.

Extension

- **Creative Task:** Students can create a story with multiple scenes, where one sprite interacts with others based on user input.
- **Discussion Prompt:** Discuss how Scratch can be used to create various types of projects such as games, stories, and simulations. How can you take your Scratch project to the next level?
- **Oral Questions:**
    - How does event handling improve the interactivity of a Scratch project?
    - Can you use Scratch to create educational games? How would you design one?
    - What happens if you don't properly use control blocks like "wait" and "forever" in a Scratch program?

Assessment and Evaluation

- **Quiz:** Multiple-choice and short-answer questions about Scratch blocks, programming concepts, and project creation.

- **Practical Test:** Students will demonstrate their Scratch skills by creating a project with motion, looks, and event blocks.
- **Peer Review:** Students will present their projects, explaining the steps they took to create them and how they used different Scratch blocks.

Suggested Activity

- **Group Activity:** Have students collaborate in creating a simple educational game that teaches a specific subject (e.g., math, spelling, geography).
- **Scratch Showdown:** Students create a project using specific Scratch blocks (e.g., a game using the "repeat" block, a dance animation using "say" blocks).

4 Cyber Security

Teaching Objectives

By the end of this lesson, students will be able to:

- Understand different types of networks (PAN, LAN, WAN) and their uses.
- Learn common internet terminologies like IP addresses, URLs, and bandwidth.
- Identify various types of cyber threats, such as malware and phishing.
- Understand how to protect personal data using antivirus software and firewalls.
- Discuss the importance of cybersecurity in preventing cybercrime.

Number of Sessions	
Theory	Practical
3	2

Teaching Plan

Engagement (Introduction)

1. **Question Prompt:** "What steps do you take to protect your personal information online? Do you know how firewalls and antivirus work?"
2. **Interactive Activity:** Show a demonstration of how a computer network functions, using examples of different network types (PAN, LAN, WAN).
3. **QAXP Activity:** Students will brainstorm the various cybersecurity risks they face daily and how they can secure their devices.

Lesson Delivery

1. **Introduction to Scratch**
   - **What is a Network?:** Explain how networks allow computers and devices to connect and share resources.

- **Types of Networks:**
  - **PAN (Personal Area Network):** For connecting personal devices like smartphones, laptops, etc.
  - **LAN (Local Area Network):** Used in schools, homes, or offices to connect devices within a small area.
  - **WAN (Wide Area Network):** Covers a broader geographic area, connecting multiple LANs across cities or countries.
- **Activity:** Demonstrate the differences between PAN, LAN, and WAN with real-life examples like home networks and the internet.

2. Common Internet Terminologies
   - **IP Address:** An identifier for each device on a network, allowing communication across the internet.
   - **URL (Uniform Resource Locator):** The address used to locate resources on the internet (websites, videos, etc.).
   - **Bandwidth:** The capacity of a network to transfer data.
   - **Activity:** Ask students to look up the IP address and URL of their favorite websites and understand how they function.

3. Cybersecurity and Malware
   - **Malware:** Malicious software designed to harm, exploit, or disable computers and networks.
   - **Types of Malware:**
     - **Viruses:** Programs that replicate and spread to other devices.
     - **Trojan Horses:** Malware disguised as legitimate software.
     - **Worms:** Malware that spreads through networks without user interaction.
   - **How to Protect Against Malware:** Use of antivirus software, regular system scans, and safe browsing practices.
   - **Activity:** Have students run a malware scan on their devices or demonstrate using an antivirus program in class.

4. Cybercrime and its Impact
   - **What is Cybercrime?:** Explain various types of cybercrimes, including identity theft, hacking, phishing, and cyberbullying.
   - **Phishing:** Fraudulent attempts to obtain sensitive information by impersonating legitimate services.
   - **Impact of Cybercrime:** Loss of personal information, financial harm, and damage to reputation.

- **Activity:** Discuss a case study on phishing scams and how they can be avoided.

5. **Tools to Protect Data**
   - **Antivirus Software:** Software used to detect and prevent malware and other security threats.
   - **Firewalls:** Security systems that monitor and control incoming and outgoing network traffic.
   - **Encryption:** The process of converting data into a secure format to protect it from unauthorized access.
   - **Activity:** Show how a firewall blocks suspicious internet traffic and explain how encryption secures online transactions.

Practical Activities

- **Activity 1:** Students will identify the types of networks used in their homes, schools, and public spaces (e.g., Wi-Fi networks).
- **Activity 2:** Using an antivirus software program, students will run a system scan and review the results, learning how to address any detected issues.
- **Group Project:** Students will design a simple cybersecurity strategy for a fictional company, including the use of firewalls, antivirus software, and encrypted communications.

Extension

- **Creative Task:** Have students create a poster on how to protect personal data online, including tips on avoiding malware, creating strong passwords, and avoiding phishing scams.
- **Discussion Prompt:** How do strong cybersecurity practices benefit individuals and organizations? How can they prevent widespread data breaches and identity theft?
- **Oral Questions:**
  - What are the differences between PAN, LAN, and WAN?
  - How does an IP address help in identifying devices on a network?
  - Why is cybersecurity important in preventing cybercrime?
  - How can you tell if a website is legitimate or a phishing attempt?
  - What role do firewalls play in network security?

Assessment and Evaluation

- **Quiz:** Multiple-choice and short-answer questions on network types, cybersecurity tools, and malware prevention.
- **Practical Test:** Students demonstrate running an antivirus scan and explain how to secure a device from malware.
- **Peer Review:** Students will assess each other's cybersecurity posters or presentations on internet safety.

Suggested Activity

- **Group Activity:** Simulate a scenario where students must protect a company's data from an external cyberattack. They will apply tools like firewalls, encryption, and safe practices to secure the system.
- **Scratch Showdown:** Create a series of short safety tips for students to share with friends and family about staying safe online.