Co-funded by
the European Union

MODULE PLAN AND CONTENT

Module 3

Introduction Coding -
Introduction to Programming

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect
those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

1. MODULE PLAN

Module Plan	
Module Title	**Introduction to Coding**
Duration	3 hours (total)
Learning Objectives	By the end of this module, participants will be able to - • Understand the basic concepts of programming and coding. • Recognize different programming languages and their uses. • Write simple programs using fundamental programming constructs (e.g., variables, loops, conditionals). • Develop problem-solving skills through coding exercises. • Gain familiarity with the overall structure of a program. • Understand the importance of algorithms and logical thinking in programming. • Learn how to debug simple code errors. • Explore real-world applications of coding in various industries. • Promote computational thinking and creativity through programming activities.
Lesson Plan	**Digital Design & Development Basics** **Training Module for Deaf Youth**

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Module Plan	
	Duration: ~60 minutes (3 topics × 20 minutes)
	1. What is Coding and Programming? (20 minutes) **Objective:** • Understand what coding and programming are and their importance in today's digital world. **Main Points to Cover:** • Definition of coding and programming • Difference between programming languages and examples (e.g., Python, JavaScript) • Real-world examples of coding applications (apps, websites, games) **Activity:** • Discuss and brainstorm where students encounter coding in everyday life • Watch a short video explaining the importance of programming in technology **Teaching Methodology:** • Lecture with visual aids (slides, videos) • Interactive discussion
	2. Basic Concepts and Building Blocks of Coding (20 minutes) **Objective:** • Introduce fundamental programming concepts like variables, data types, and basic control structures. **Main Points to Cover:**

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Module Plan	
	• Variables and data types (numbers, text) • Basic operators and expressions • Simple control flow: if-else statements and loops • The concept of algorithms and problem-solving steps **Activity:** • Hands-on exercise: create simple pseudocode or flowcharts for everyday tasks (e.g., making a sandwich, getting ready for school) **Teaching Methodology:** • Demonstration with live coding/examples • Guided practice exercises, group work **3. Writing Your First Program (20 minutes)** **Objective:** • Enable students to write and run their first simple program. **Main Points to Cover:** • Introduction to a beginner-friendly programming environment (e.g., Scratch, Code.org, Python IDLE) • Structure of a simple program (print statement, input, basic logic) • Debugging common errors **Activity:** • Step-by-step coding: Students write a program that asks for their name and greets them • Peer review and debugging exercises **Teaching Methodology:** • Live coding demonstration

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Module Plan	
	• Guided practice with support and feedback • Peer collaboration
Materials Needed	**1. Visual Aids and Presentations:** • Slides or PowerPoint presentation with key concepts • Short videos explaining coding and programming (optional) **2. Computers and Software:** • Computers or laptops for each student or pairs • A beginner-friendly programming environment (e.g., Scratch, Python IDLE, Code.org, Blockly) **3. Handouts and Worksheets:** • Pseudocode or flowchart templates for activities • Worksheets with exercises on basic concepts and problem-solving • Cheat sheets for common programming syntax and commands **4. Interactive Tools:** • Online coding platforms or websites for practice (e.g., Code.org, Khan Academy, Trinket) • Debugging exercises or game-like coding challenges **5. Additional Materials:** • Whiteboard and markers for explanations and discussions • Printed examples of simple programs and algorithms • Notebooks or paper for scratch work or brainstorming

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

2. MODULE CONTENT

Content	Description	Content
Lesson Content	**Digital Literacy**	**Digital literacy** means having the knowledge and skills to use digital tools like design software and programming languages to create apps, websites, and online content. It includes understanding how digital systems work and being able to build and design simple digital products. **Key Digital Terms:** • **Programming Language** – A language used to write instructions that a computer can understand. • **Code** – The set of instructions written in a programming language. • **Algorithm** – A step-by-step set of instructions to solve a problem. • **Variable** – A storage location in a program that holds data. • **Data Type** – The kind of data (e.g., integers, strings, floats) stored in a variable. • **Function / Method** – A block of code that performs a specific task and can be reused. • **Loop** – A sequence of instructions that repeats until a condition is met. • **Condition / If-Else** – A statement that performs different actions based on whether a condition is true or false. • **Debugging** – The process of finding and fixing errors or bugs in code. • **Syntax** – The rules that define the structure of code in a programming language. • **Input** – Data that a program receives from the user or another source.

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Content	Description	Content
		• **Output** – Data produced by the program for the user. • **IDE (Integrated Development Environment)** – A software application that provides tools to write, test, and debug code. • **Bytecode** – An intermediate code that can be interpreted or compiled. • **Compilation** – The process of translating code into a machine language that a computer can execute. • **Machine Language** – The binary code that a computer's processor understands directly. • **Comments** – Notes written in code to explain what the code does; ignored by the computer. • **Syntax Error** – An error due to incorrect code that violates the language's rules. • **Logic Error** – An error where the code runs but produces incorrect results.
Video summary	**Youtube Links**	1. **"What is Coding?" – CrashCourse Computer Science** https://www.youtube.com/watch?v=fikvFQoYZ3Q An easy-to-understand explanation of what programming is and why it's important. 2. **"Introduction to Programming" – freeCodeCamp** https://www.youtube.com/watch?v=8oRjP8yj_w8 A beginner-friendly overview of programming fundamentals. 3. **"Learn Programming in 1 Hour" – Tech with Tim** https://www.youtube.com/watch?v=BoardJz_VRc A quick summary of core programming concepts designed for beginners.

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130

Content	Description	Content
		4. **"What is a Programming Language?" – Computer Science Gaming** https://www.youtube.com/watch?v=7sA8vfQWWvg Short explanation of programming languages and how they work.
Activity		**1. Group Activity:** **Objective:** • Encourage students to apply basic programming concepts such as sequences, conditions, and loops by designing a simple, step-by-step algorithm for a real-world task. **Activity Description:** 1. Divide students into small groups (3-4 students per group). 2. Each group chooses or is assigned a simple everyday task (e.g., making a sandwich, brushing teeth, getting ready for school). 3. Students work together to create a detailed step-by-step algorithm or flowchart describing the task, including decision points (if-else conditions) and repetitive steps (loops). 4. Each group presents their algorithm to the class, explaining how it works with emphasis on the logical steps and decision points. **Materials Needed:** • Large sheets of paper or whiteboards, markers, or printed templates for flowcharts.

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Content	Description	Content
		• Handouts with basic algorithm symbols and flowchart guidelines (optional). **Learning Outcome:** • Students will demonstrate understanding of programming logic and how real-world tasks can be broken down into code-like instructions. • They will develop problem-solving and collaboration skills.
		2. Role-Play: **Objective:** • To help students understand how to explain programming concepts clearly and to simulate real-world communication in coding. It also reinforces their understanding of programming steps and decision-making. **Activity Description:** 1. Students pair up; one student acts as the **"Coding Coach"**, and the other as the **"Beginner"**who wants to learn how to program a simple task (e.g., making a sandwich, brushing teeth). 2. The **"Beginner"** asks the **"Coding Coach"** to explain how to complete the task through a step-by-step "coding" process. 3. The **"Coding Coach"** must describe the process in simple, clear instructions, including decision points and repetitions, as if teaching a beginner how to think like a programmer.

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130

Content	Description	Content
		4. Switch roles after completing the explanation. 5. After both roles are played, students discuss the importance of clear instructions and logical thinking in programming. **Materials Needed:** • None, just space for role-play. • Optionally, provide students with example tasks or steps to help them start. **Learning Outcome:** • Students will practice translating real-world tasks into logical, step-by-step instructions. • They will develop skills in explaining technical concepts clearly and listening actively.
		3. Real-Life Scenarios: **Objective:** • To help students understand how to break down a daily routine into clear, logical steps similar to coding. This activity promotes problem-solving, sequencing, and decision-making skills. **Activity Description:** 1. Pair up students; one student plays the **"Programmer"** (the person designing the instructions), and the other is the **"User"** (the person following the instructions).

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Content	Description	Content
		2. The **"Programmer"** is asked to describe a typical morning routine (e.g., "Getting ready for school") as if they are creating a simple "program" or set of instructions to follow. o They should include steps like waking up, brushing teeth, choosing clothes, eating breakfast, etc. o They should incorporate decision points (e.g., "if it is cold, wear a jacket") and repetitions (e.g., "brush teeth twice"). 3. The **"User"** listens and then performs the routine exactly as described. 4. Switch roles so that both students get to practice giving instructions and following them. 5. Afterward, discuss what was easy or difficult about understanding and following instructions, and how this relates to breaking problems into steps in programming. **Materials Needed:** • No specific materials needed, just space for role-play. • Optional: Provide a checklist or example routine to help students prepare. **Learning Outcome:** • Students will experience how to translate everyday routines into a sequence of instructions, similar to programming. • They will improve their communication skills and understanding of conditional logic and sequencing.

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Content	Description	Content
Assessment		**1. What is a Program ?** A) A type of hardware used in computers. B) A game played on a computer. C) A set of instructions that a computer follows to perform a task. ✅ D) A device that stores data.
	Quiz	**2. Which of the following is an example of a variable in programming?** A) A command to print text on the screen. B) A storage location that holds data, like a number or text. ✅ C) A message that explains what the code does. D) A type of programming language.
		3. What does a loop do in a program? A) Checks if a condition is true or false. B) Repeats a set of instructions multiple times. ✅ C) Stores data for later use. D) Sends data to another computer.

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Content	Description	Content
		4. Why is debugging important in coding? A) To add new features to the program. B) To identify and fix errors or bugs in the code. ✅ C) To make the program run faster. D) To translate code into different languages.

PROJECT NO: 2023-2-IT03-KA220-YOU-000179130
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.