Co-funded by
the European Union

MODULE PLAN AND CONTENT

Module 9

Java / Javascript

Version 1.0

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	**Java and Javascript**
Duration	3 hours (total)
Learning Objectives	By the end of this module, participants will be able to: • Understand the purpose and use of Java and JavaScript. • Recognize the differences and similarities between the two languages. • Write and run simple programs in Java (console-based) and JavaScript (browser-based). • Apply basic programming logic, variables, input/output, and conditions.
Lesson Plan	**Digital Design & Development Basics** **Training Module for Deaf Youth** **Duration:** ~180 minutes (3 topics × 50 minutes + 2 breaks × 10 minutes)
	1. What Are Java and JavaScript? (50 minutes) **Objective:** Introduce both languages and explain where and how they are used. **Main Points to Cover:** • What is Java? What is JavaScript?

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	
	• Where are they used? (apps, web, games, etc.) • Main differences (compiled vs interpreted, server vs browser) • Web-based environments: Replit for Java, JSFiddle for JavaScript • Examples of applications made with each language. **Activity: "Code Guess"** • Participants are shown 5 short code snippets and asked to identify whether they are Java or JavaScript and explain why. **Teaching Methodology:** • Interactive presentation with visual code examples. • Small group discussion. • Fill-in worksheet (match purpose to language).
	2. Let's Code in Java (50 minutes) **Objective:** Write and run simple Java programs using online compilers with no installation required. **Main Points to Cover:** • Structure of a Java program (class, main() method). • Printing output with System.out.println(). • Variables and data types (int, String). • Input using Scanner.

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	
	• Simple logic (if statements, loops). **Activity:** • Open a pre-made online tool link with starter code • Modify it to display a greeting based on a name or age • Bonus: Add a calculation (e.g., age in 10 years) **Teaching Methodology:** • Online Java Editor (e.g. jdoodle.com) • Instructor-provided starter link (with template) • Code-along with instructor • Group support for testing and debugging • Minimal code, max logic focus **3. Javascript (50 minutes)** **Objective:** Write browser-based JavaScript programs with live output using online editors. **Main Points to Cover:** • JavaScript basics: variables (let), DOM manipulation • HTML + JS integration • Event handling (onclick) **Activity:**

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	
	• Open instructor's online javascript template • Add a counter to a button that increases each time it's clicked • Add message when reaching 5 clicks **Teaching Methodology:** • Online Javascript Editor (e.g. jsfiddle.net) • Pre-made HTML + JavaScript template • Code-along + play-based exploration • Immediate visual feedback • Group reflection and sharing
Materials Needed	▪ Internet-connected desktop, laptops or tablets ▪ Access to: o Online Java Editor (e.g. jdoodle.com) o Online Javascript Editor (e.g. jsfiddle.net)

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	**Key Digital Terms:**	**Key Digital Terms:** □ **Variable (Java and JavaScript)**: A named storage location for data, like a number or a word. □ **Data Type (Java and JavaScript)**: Defines what kind of data a variable holds (e.g., number, text, true/false). □ **Function / Method (Java and JavaScript)**: A block of reusable code that performs a task. □ **Class (Java)**: A blueprint for creating objects, defining properties and behaviors. □ **Object (Java and JavaScript)**: A collection of properties and methods based on a class. □ **Loop (Java and JavaScript)**: Repeats code multiple times (e.g., for, while loops). □ **Condition / If-Else (Java and JavaScript)**: Allows code to make decisions depending on certain criteria. □ **Array (Java and JavaScript)**: A list-like structure that holds multiple values in a single variable. □ **String (Java and JavaScript)**: A data type that represents text (e.g., "Hello world"). □ **Console (Java and JavaScript)**: A space where output or debugging messages appear.

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
		□ **Input (Java)**: Getting data from the user using tools like the Scanner class. □ **Output (Java and JavaScript)**: Displaying results using System.out.println() in Java or console.log() in JavaScript. □ **DOM (JavaScript)**: A structured representation of an HTML page that JavaScript can read and modify. □ **Event Listener (JavaScript)**: A function that runs in response to user actions like clicks or key presses. □ **OnClick (JavaScript)**: A specific event that happens when a user clicks an element. □ **Compiler (Java)**: A tool that translates code into bytecode before execution. □ **Interpreter (JavaScript)**: A system that reads and runs code line-by-line directly in the browser. □ **Syntax (Java and JavaScript)**: The rules and structure for writing valid code in a programming language.
Video summary	**Youtube Links**	• **Algorithm** – https://www.youtube.com/watch?v=Ld_iFKCxeS8& • **Backend Development** – https://www.youtube.com/watch?v=2y9eJw0ZU4s **Coding** – https://www.youtube.com/watch?v=FodVPon1Nbk • **Control Structure -** https://www.youtube.com/watch?v=BFpcRznhvCI • **Frontend Development -** https://www.youtube.com/watch?v=cCNqo7lzhPQ • **Java -** https://www.youtube.com/watch?v=APW6WP1Q1Tg • **Javascript -** https://www.youtube.com/watch?v=mEUPxfzEgFE

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
		• **Loop** - https://www.youtube.com/watch?v=9S745R-Xr3k • **Variable** - https://www.youtube.com/watch?v=Mc_Msxqdlks
Activity		**1. Group Activity: One Task, Two Languages** **Goal:** Learn how the same programming task can be completed using both Java and JavaScript, and understand the similarities and differences between the two languages. Participants work in small teams. Each team gets a simple coding task: **"Create a personalized greeting program that asks for a name and says hello."** In relay style, team members take turns writing one part of the program: • First, in **Java** using jdoodle.com • Then, in **JavaScript** using JSFiddle They follow these steps: • One member sets up the structure (e.g., class in Java or HTML in JS) • Another handles input (e.g., Scanner in Java or input element in JS) • Another manages output (e.g., System.out.println() or DOM update) Each team presents their two versions, explains their approach, and reflects on: • What parts were similar? • What was harder or easier?

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
		• Which environment felt more intuitive?
		2. Role-Play: Code Collaboration **Goal:** Practice teamwork, basic programming logic, and real-world communication in coding tasks using Java and JavaScript. Each team simulates a coding situation where members must: • Plan a small program together • Divide the task between Java and JavaScript roles • Communicate their steps clearly to each other **Scenario Examples:** • Building a greeting program together (one in Java, one in JavaScript) • One team member writes logic, another handles display • One team debugs code while another explains the logic **Roles:** • **Developer (Java)** – explains and writes the Java version of the task • **Developer (JavaScript)** – explains and writes the JavaScript version • **Reviewer** – gives feedback on clarity, structure, and output • **Presenter** – summarizes what the team did and why After acting it out, teams discuss:

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
		• Was the communication between Java and JavaScript developers clear? • Did everyone understand both versions of the task? • What could be improved for future collaboration?

		3. Real-Life Scenarios:
		Goal:
		Apply programming logic and basic coding skills in Java and JavaScript to real-life inspired mini-projects.
		Participants receive individual or pair-based tasks such as:
		• "Create a welcome message program for a new website visitor"
		• "Build a basic calculator that adds two numbers"
		Each task requires:
		• A **plan** written as an algorithm (step-by-step logic)
		• Use of one or more **online coding tools** (jdoodle.com for Java, JSFiddle for JavaScript)
		• A short **reflection**, answering:
		  ○ What helped you complete this task?
		  ○ What language felt more natural for the problem?
		  ○ What was difficult or unexpected?
		Participants share their code and discuss differences in how Java and JavaScript handle similar logic.

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	Quiz	**1. Which tool lets you write and run Java code in the browser?** A) Eclipse B) VS Code C) JDoodle ☑ D) JSFiddle
		2. What does let clicks = 0; do in JavaScript? A) Sets a class B) Declares a variable ☑ C) Prints output D) Loads HTML
		3. Which line correctly prints text to the console in Java? A) Console.Write("Hello"); B) print("Hello"); C) System.out.println("Hello"); ☑ D) echo("Hello");

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. Which platform is better for writing code that changes a web page? A) Java B) JavaScript ☑ C) C++ D) Python

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.