Week 1: Course Introduction & Tools Overview

Developer Tools: VSCode & Marp

- **Visual Studio Code (VSCode)** – a lightweight **IDE** with rich features (file explorer, text editor, version control integration, terminal, etc.). It's extensible via plugins and serves as the primary coding environment.
  - **2025 Stats:** Most popular IDE with 60,000+ extensions available
  - Supports all major programming languages with IntelliSense
- **Marp** – a toolchain to create slides with Markdown. *Write* content in a Markdown file and Marp **compiles it into presentation formats** (PDF, PPT, HTML). Using the Marp VSCode extension, you can preview slides and export them easily.
- These tools streamline our workflow: VSCode for all-in-one development and Marp for documentation and presentations.

Git & GitHub Basics

- **Git** is a distributed version control system for tracking code changes. It runs locally and lets you save versions of your code over time.
    - **2025 Adoption:** 93.87% of developers use Git (up from 87.1% in 2016)
- **GitHub** is an **online platform for Git repositories**, providing cloud storage for code and facilitating collaboration. It acts as a portfolio for projects and a central hub to share code with teammates.
- In this course, you'll use GitHub to access course materials and submit assignments. Common ASE course material is available in the public GitHub repo (e.g. the `nkuase/ASE` repo) for reference.

Course Overview & Policies

- **Course Introduction:** The course covers *high-quality software engineering* practices, including development processes, design principles, and team collaboration. We will manage a project through all stages from requirements to testing.
- **Course Policies:** Attendance and participation are crucial. Assignments must be submitted on time; late work may receive little or no credit. **Academic integrity** is emphasized – all code must be your own work (plagiarism or unauthorized collaboration is prohibited).
- **Tools & Workflow:** We will use VSCode, Git/GitHub for version control, and Marp for documentation. Use these consistently to document your progress and collaborate.

High-Quality Software: Storytelling & Assignments

- **Software as Storytelling:** Writing code and documentation is akin to storytelling. Your code should be clear and well-structured so that other developers (and your future self) can follow the "story" easily. Good software design tells a coherent story about the problem and solution.
- **Communication:** Throughout the course, you'll practice explaining your design decisions. This includes writing user stories, use cases, and documentation that communicate the purpose and usage of your software.
- **Assignments Overview:** The course assignments are structured to apply concepts in practice. Early assignments focus on individual skills (tools setup, simple design tasks). Later assignments build toward a **capstone project**, where you'll work in teams to deliver a software product incrementally. Each assignment reinforces topics from lectures (e.g. version control, requirements, design, testing).