Oct 24, 2025

Class 4 (Intro to C# and .NET)

Class records 🎬 Class 4 (Intro to C# and .NET)

Summary

Amospikins provided a comprehensive overview of backend essentials, explaining C# as a powerful, simplified language created by Microsoft, and detailing the evolution of the .NET framework into an open-source, cross-platform technology. They clarified the distinction between frameworks and libraries in response to Nelly Blues' question, with Michael Ikhimi adding a framework can be seen as a collection of libraries, and discussed the advantages of statically typed languages like C# over dynamically typed ones. Amospikins also assessed the value and scarcity of programming languages in the job market, emphasizing the versatility of .NET for various applications and recommending Visual Studio Code for cross-platform compatibility, with Anthony Sopuruchukwu Ubogu confirming the next session will focus on installations.

Details

Notes Length: Standard

- **Backend Essentials** Amospikins defined backend as a blend of programming languages and databases, noting that programming languages run directly on the server, which they clarified is essentially a computer. They explained that any language capable of running directly on a system can be used for backend development.
- **C# Programming Language** Amospikins detailed that C# was created to compete with Java and simplify memory management compared to C++. They highlighted that C# combines the power of C++ with greater simplicity in writing

code, making it a robust yet easier-to-use language. Amospikins also mentioned that Microsoft created C#.

- **.NET Framework Evolution** Amospikins explained that the .NET framework was initially developed to run C# and other Microsoft languages like VB.NET on Windows systems. They then discussed the introduction of .NET Core, which allowed Microsoft technologies to run cross-platform on Mac and Linux. Eventually, both were unified into a single .NET platform that is now open-source, enabling broader use and community contributions.
- **Versatility of .NET** Amospikins emphasized the wide range of applications possible with .NET, including backend development, desktop applications, mobile apps using technologies like MUI, game development with Unity, and embedded systems. They stressed the importance of learning C# within the .NET ecosystem due to its extensive capabilities.
- **Framework vs. Library** Amospikins clarified the difference between frameworks and libraries in response to Nelly Blues' question. They explained that a library, like React, is a collection of tools for a specific function, while a framework, like .NET, combines various technologies to manage broader server-side tasks, with Michael Ikhimi suggesting a framework can be seen as a collection of libraries.
- **Statically vs. Dynamically Typed Languages** Amospikins explained that statically typed languages check for data type errors at compile time, leading to earlier error detection and more reliable code. Conversely, dynamically typed languages check types at runtime, which can result in unexpected crashes and performance overhead. They noted that C# is statically typed, while Python and JavaScript are dynamically typed.
- **Value and Scarcity in Programming Languages** Amospikins discussed how the financial relevance of a programming language depends on its value and scarcity, noting that older languages like C, despite being "old," maintain high demand for maintaining legacy systems. They also highlighted that while NodeJS and TypeScript are popular globally, Java and .NET are highly valued in Nigeria's core financial sector. Amospikins also explained that personally, they value knowing multiple languages to ensure continuous work.
- **Checking .NET Installation** Amospikins demonstrated how to check for .NET SDK and runtime installation on a system using the command prompt `dotnet –info`. They also noted that for running C# code, the file extension must be `.cs`,

and unlike NodeJS, C# requires several installations, which Microsoft simplifies with an IDE.

- **Integrated Development Environments (IDEs)** Amospikins discussed different IDEs for coding, specifically Visual Studio and Visual Studio Code. They highlighted that Visual Studio is a larger, more comprehensive IDE (around 8 GB) compared to Visual Studio Code (around 100 MB), which is a lighter, cross-platform version. Visual Studio offers a more integrated experience for C# and machine learning without needing separate installations, while VS Code requires individual component installations.
- **Cross-Platform Compatibility** Amospikins proposed using Visual Studio Code initially for cross-platform compatibility, especially since Visual Studio is not available on Mac and Linux. They also suggested that for the next class, they would use their Linux system to ensure everyone, regardless of their operating system, could follow along with the installation process.
- **Upcoming Session and Installations** Anthony Sopuruchukwu Ubogu clarified that the next class would focus on installations. Amospikins confirmed this, stating that the next session would involve demonstrating the installation process, possibly using a Linux environment to accommodate all attendees.