Turiba University

PARTHEEV PULIYANMANAYIL RADHAKRISHNAN

HISTORY OF PROGRAMMING

PROFESSIONAL BACHELOR THESIS

Study Programme: Computer Systems

Author: **PARTHEEV**

***Thesis* Advisor:** **Janis Peksa**

Riga, 2025

What Is a Programming Language?

- A program is a sequence of instructions and decisions executed by a computer.
- Programming languages are tools that allow humans to communicate with computers.
- They evolved to make programming easier, faster, and less error-prone.
- Programs control hardware and perform tasks from operating systems to web browsers.

The Machine and Assembly Era (1940s–1950s)

- **Machine language:** first generation; written in binary (0s and 1s).
  Example: 1101 0000 0000 0111
- **Assembly language:** second generation; used mnemonics like ADD, MOV, JMP.
- Required **deep hardware knowledge** and manual memory management.
- Each CPU type had its own instruction set — programs weren't portable.

The Rise of High-Level Languages (1950s-1970s)

- Third generation languages: more abstract and closer to English.
- Examples: **FORTRAN (1957)** – for scientific computing; **COBOL (1959)** – for business; **C (1972)** – for system software.
- Introduced **structured programming**, **loops**, and **functions/methods**.
- Programs became **portable** and **easier to maintain**.

Object-Oriented and Fourth-Generation Languages (1980s–1990s)

- **Object-Oriented Programming (OOP):** data + behavior = objects.
- Popular OOP languages: **C++**, **Java**, **Python**.
- Fourth-generation languages (4GLs): emphasized *what* to do, not *how* (e.g., **SQL**, **SAS**, **Mathematica**).
- Designed for **productivity**, **reusability**, and **faster development**.

The Internet and Cross-Platform Needs (1990s)

- **Platform Independence: Java (1995)** used the Java Virtual Machine (JVM) to achieve its goal of "Write Once, Run Anywhere," becoming the standard for large-scale enterprise server applications.

- **Client-Side Scripting: JavaScript (1995)** was initially designed to run simple scripts inside the web browser. It evolved rapidly to become the primary language for interactive front-end development and later, server-side (Node.js).
- **Readability Focus: Python (1991)** gained momentum due to its clean syntax and readability, growing from a scripting tool into the primary language for scientific computing and data analysis.

Modern Trends and Future Directions (2000s–Today)

- Rise of **multi-paradigm** languages (e.g., Python, JavaScript).
- Emphasis on **readability**, **automation**, and **AI integration**.
- **Open-source** ecosystems and cloud computing expanded programming's reach.
- Emerging areas: **functional programming**, **low-code tools**, and **AI-assisted coding**.