

Turiba University

KARTIK JYOTI

**HISTORY OF PROGRAMMING
PROFESSIONAL BACHELOR THESIS**

computer systems

Author:

KARTIK JYOTI

Thesis Advisor:

(Jānis Pekša Dean of IT

Riga, 2025

HISTORY OF PROGRAMMING

1. Early Beginnings (1800s–1940s)

The story of programming starts way back in the 1800s. Charles Babbage designed a special machine called the *Analytical Engine*, and Ada Lovelace wrote the first set of instructions for it — which is why she's known as the world's first programmer. Later, in the 1930s and 40s, early computers like ENIAC used punched cards to run basic programs.

2. Machine and Assembly Languages (1940s–1950s)

In the beginning, computers only understood machine code, which is basically just 0s and 1s. It was hard for humans to work with, so assembly language was introduced. Instead of binary, programmers could now use short words like ADD or MOV, making coding much easier.

3. High-Level Languages (1950s–1970s)

This period changed everything. Programming started becoming more human-friendly.

Languages like:

- FORTRAN → used for scientific work
- COBOL → used in business
- LISP → used in early AI
- BASIC → made learning programming simpler

These languages were closer to English, so writing programs became much more understandable.

4. Structured and System Languages (1970s–1990s)

This era introduced powerful and well-organized programming styles.

The most important language here is C (1972), which became the base for many future languages and operating systems like UNIX.

Pascal encouraged clean, structured code, and C++ (1985) introduced object-oriented programming, helping developers build larger and more organized projects.

5. Modern Languages (1990s–2010s)

Once the internet took off, new languages were created to match the growing digital world.

- Java brought the idea of “write once, run anywhere”
- Python became popular for being simple and beginner-friendly
- JavaScript made websites interactive
- PHP powered many early websites
- C# was developed by Microsoft for Windows software

These languages helped build websites, mobile apps, and modern software.

6. Current Trends (2010s–Present)

Programming today is faster, cleaner, and more advanced than ever.

- Swift for iOS apps
- Kotlin for Android
- Go and Rust for performance and security
- Python and R for AI and data science
- Tools like low-code and no-code allow people to build apps without deep coding skills

Modern programming focuses on speed, simplicity, and smarter technologies.

Conclusion

The history of programming shows how far we've come—from simple mechanical ideas to advanced software powering our daily lives. Each new generation of programming languages made development easier and more powerful. As technology keeps growing with AI, cloud computing, and automation, programming will continue evolving and shaping the future.