

# Lecture 09

## Ada to Software Engineering

**Mr. Mubashir Ali**

**Lecturer (Dept. of Computer Science)**

[dr.mubashirali1@gmail.com](mailto:dr.mubashirali1@gmail.com)

# Summary of Previous Lecture

- 1. ALGOL 68**
- 2. COBOL 60**
- 3. PL/1**
- 4. BASIC**
- 5. Early Dynamic Languages**
- 6. Simula 67**
- 7. Pascal**
- 8. C**
- 9. Prolog**

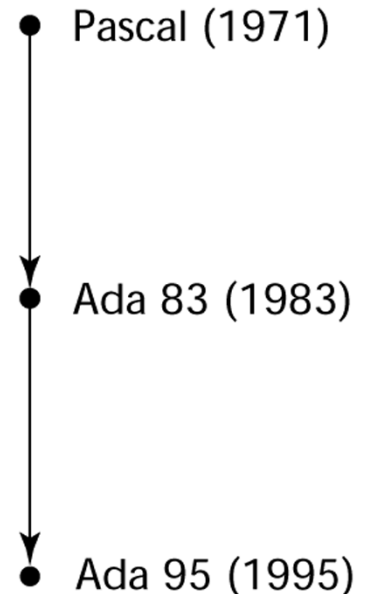
# Outline

- 1. Ada**
- 2. Smalltalk**
- 3. C++**
- 4. Java**
- 5. C#**
- 6. Scripting Language for Web**
- 7. Programming Language Evolution**
- 8. Software Engineering**

# Ada – 1983

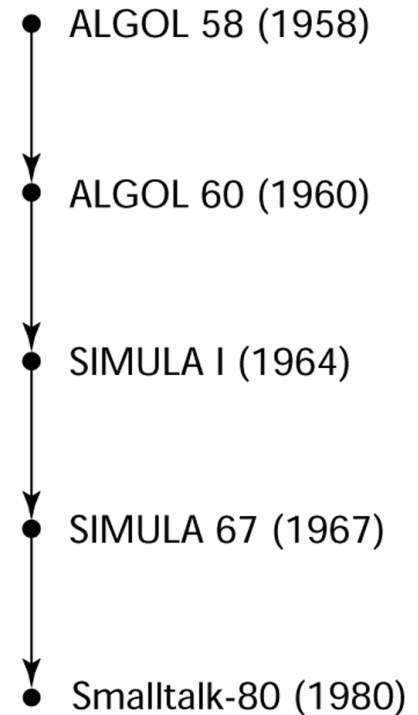
## History's largest design effort

- **Huge design effort, involving hundreds of people, lots of money and about eight years of designing effort**
- **Contributions:**
  - **Packages - support for data abstraction**
  - **Elaborate exception handling mechanism**
  - **Facility for generic program units (templates)**
  - **Facility for concurrency - through the tasking model**
- **Comments:**
  - **Competitive design**
  - **Included all that was known about software engineering and language design at that time**
  - **First compilers were very difficult;**
    - **The first usable compiler came nearly five years after the language design was completed**



# Smalltalk - 1980

- **Developed at Xerox PARC, initially by Alan Kay, later by Adele Goldberg**
- **First full implementation of an object-oriented language**
  - It had data abstraction, inheritance, and dynamic type binding
- **Purest object-oriented language yet!**
- **Pioneered the graphical user interface everyone now uses**



# C++ - 1985

- **Developed at Bell Labs by Stroustrup**
- **Evolved from C and SIMULA 67**
- **It has facilities for object-oriented programming, taken partially from SIMULA 67 and were added to C**
- **Also has exception handling**
- **Its a large and complex language, in part because it supports both procedural and OO programming**
- **Rapidly grew in popularity, along with OOP**
- **ANSI standard approved in November, 1997**

# Java - 1995

- **Developed at Sun in the early 1990s**
- **Based on C++**
- **Significantly simplified version of C++**
- **Supports only OOP**
- **Eliminated multiple inheritance, pointers, structs, enum types, operator overloading, goto statements**
- **Includes support for applets and a form of concurrency**

# C#- 2002

- **Part of the .NET framework**
- **Language for component-based software development**
- **Based upon C++ and Java**
- **Took some ideas from Visual Basic**
- **Intermediate Language (IL)**
- **Just-in-time compiler**
- **Brought back pointers, structs, enum types, operator overloading, goto statement**
- **Has safer enum types, more useful struct types, modified switch statement**
- **Some other additions**



# Scripting Languages for Web

- Address the need for computations associated with HTML documents
- JavaScript is used on the client side and PHP is used on the server side
- JavaScript – client side
  - Primary objective is to create dynamic HTML documents and check validity of input forms
  - It is usually embedded in an HTML document
  - Its not related to Java

# Scripting Languages for Web

- PHP (Personal Home Page) – server side
  - Interpreted on the Web Server when the HTML document in which it is embedded is requested by the browser
  - Often produces HTML code as an output
  - Similar to JavaScript
  - Allows simple access to HTML form data – makes form processing easier
  - It provides support for many different database management systems and provides web access to databases

# Programming Language Evolution

- 50's was the time of discovery and description of programming language concepts
  - We had empirical approach
  - Programming languages were regarded solely as tools for facilitating the specification of programs rather than as interesting objects of study in their own right

# Discovery and Description

- Development of symbolic assembly language, macro-assembly, FORTRAN, Algol 60, COBOL and Lisp
- Discovery of many basic implementation techniques
  - Symbol table construction and look-up
  - Stack algorithms for evaluating arithmetic expressions
  - Activation record stack
  - Marking algorithms for garbage collection

# Analysis and Elaboration– 60's

- Analysis for the purpose of constructing models and theories of programming languages
- PL/1, Simula, Algol 68, Snobol
- Elaboration of earlier languages
- An attempt to achieve greater richness by synthesis of existing features and generalization
- Resulted in greater complexity
- Formal languages and automata theory with applications to parsing and compiler theory
- Theory of operational and mathematical semantics were developed
- Language definition techniques were developed
- Basic ideas of program correctness and verification
- Lambda Calculus for abstract structures

# Programming Language Evolution

- Effective software technology – 70's
  - Software Engineering Support – structured design
- 1980's – Support for SE continued
- 1990's and 2000's
  - Support for OOP and Internet
- 2010's – Aspect Oriented Programming?

# Software Engineering – 70's

- From pure research to practical management of the environment
- Decreasing hardware cost and increasing software cost resulted in breaking complexity barrier
- Development of tools and methodologies for controlling the complexities, cost and reliability of large programs
- Structured programming, modular design and verification
- Verifiable languages – Pascal, Modula

# Summary

- ✓ **Ada**
- ✓ **Smalltalk**
- ✓ **C++**
- ✓ **Java**
- ✓ **C#**
- ✓ **Scripting Language for Web**
- ✓ **Programming Language Evolution**
- ✓ **Software Engineering**