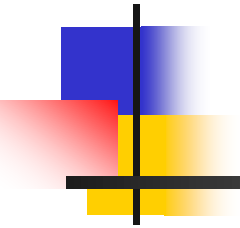
Evolution of the Major Programming Languages



Lecture 03

Instructor: C. Pu (Ph.D., Assistant Professor)

puc@marshall.edu

The Beginnings of Timesharing: BASIC

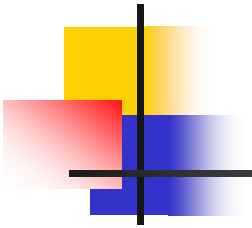
- BASIC is another programming language that has enjoyed widespread use but has gotten little respect
- BASIC was very popular on microcomputers in the late 1970s and early1980s
  - It was easy for beginners to learn, especially those who were not science oriented, and its smaller dialects can be implemented on computers with very small memories
- When the capabilities of microcomputers grew and other languages were implemented, the use of BASIC waned
- A strong resurgence in the use of BASIC began with the appearance of Visual Basic (Microsoft, 1991) in the early 1990s

BASIC Design Process

- BASIC (**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode) was originally designed at Dartmouth College (now Dartmouth University) in New Hampshire by two mathematicians, John Kemeny and Thomas Kurtz.

- Dartmouth designed a new language especially for liberal arts students.

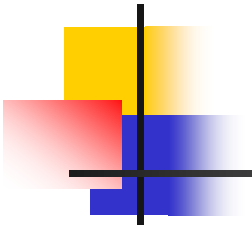
- The goals of the system were as follows
  - It must be easy for non-science students to learn and use
  - It must be "pleasant and friendly."
  - It must provide fast turnaround for homework
  - It must allow free and private access
  - It must consider user time more important than computer time



BASIC Overview

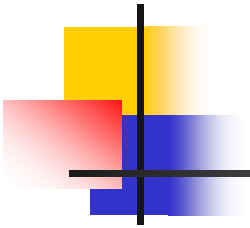
- The original version of BASIC was very small and, oddly, was not interactive:
  - There was no way for an executing program to get input data from the user
  - Programs were typed in, compiled, and run, in a sort of batch-oriented way
  - The original BASIC had only 14 different statement types and a single data type—floating-point
  - Overall, it was a very limited language, though quite easy to learn

An Imperative-Based Object-Oriented Language: Java



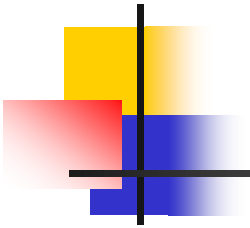
- Java's designers started with C++, removed some constructs, changed some, and added a few others.

- The resulting language provides much of the power and flexibility of C++, but in a smaller, simpler, and safer language.



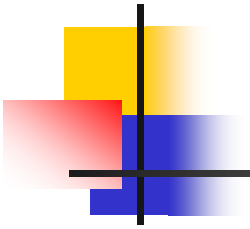
Java Design Process

- Java, like many programming languages, was designed for an application for which there appeared to be no satisfactory existing language
- In 1990, Sun Microsystems determined there was a need for a programming language for embedded consumer electronic devices, such as toasters, microwave ovens, and interactive TV systems
- ***Reliability*** was one of the primary goals for such a language
- It was also believed that neither C nor C++ provided the necessary level of reliability
- So, a new language, later named Java, was designed



Java Overview

- Java is based on C++ but it was specifically designed to be *smaller*, *simpler*, and *more reliable*
- Like C++, Java has both classes and primitive types
- Java arrays are instances of a predefined class, whereas in C++ they are not
- Java does not have pointers
- Java has a primitive Boolean type named *boolean*, used mainly for the control expressions of its control statements
- Unlike C and C++, arithmetic expression cannot be used for control expressions.



Java Overview

- All Java subprograms are methods and are defined in classes
  - Methods can be called through a class or object only
- Java supports only single inheritance of classes
- Java includes a relatively simple form of concurrency control through its synchronized modifier, which can appear on methods and blocks
- Java does not support *struct* and *union*.
- Java uses implicit storage deallocation for its objects, often called garbage collection
- Java includes assignment type coercions (implicit type conversions) only if they are widening (from a "smaller" type to a "larger" type).

Scripting Languages: JavaScript

- Use of the Web exploded in the mid-1990s after the first graphical browsers appeared
- The need for computation associated with HTML documents, which by themselves are completely static, quickly became critical
- Computation on the server side was made possible, which allowed HTML documents to request the execution of programs on the server, with the results of such computations returned to the browser in the form of HTML documents
- Computation on the browser end became available with the advent of *Java applets*

Scripting Languages: JavaScript

- JavaScript (Flanagan, 2002) was originally developed by Brendan Eich at Netscape.
  - Its original name was Mocha.
  - It was later renamed LiveScript.
- In late 1995, LiveScript became a joint venture of Netscape and Sun Microsystems and its name was changed to JavaScript.
- JavaScript has gone through extensive evolution, moving from version 1.0 to version 1.5 by adding many new features and capabilities.
- A language standard for JavaScript was developed in the late 1990s

Scripting Languages: JavaScript

- Although a JavaScript interpreter could be embedded in many different applications, its most common use is embedded in Web browsers.
- JavaScript code is embedded in HTML documents and interpreted by the browser when the documents are displayed.
- The primary uses of JavaScript in Web programming are to validate form input data and create dynamic HTML documents.
- JavaScript also is now used with the Rails Web development framework.

Scripting Languages: JavaScript

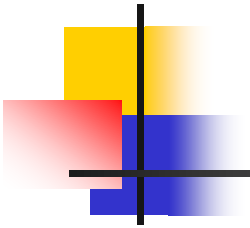
- In spite of its name, JavaScript is related to Java only through the use of similar syntax.
- Java is strongly typed, but JavaScript is dynamically typed.
- JavaScript's character strings and its arrays have dynamic length.
  - Array indices are not checked for validity, although this is required in Java.

Programming Language Implementation

- Two of the primary components of a computer are its *internal memory* and its *processor*.
  - *Internal memory*: store programs and data.
  - *Processor*: a collection of circuits that provides a realization of a set of primitive operations, or machine instructions.

- The *machine language* of the computer is *a set of instructions*.
  - The ***only language*** that most hardware computers "understand".

- Theoretically, a computer could be designed and built with a particular high-level language as its machine language.
  - ***But, it would be very complex and expensive.***
  - ***But, it would be highly inflexible.***



Programming Language Implementation

- *The more practical machine design choice implements in hardware a very low-level language that provides the most commonly needed primitive operations and requires system software to create an interface to programs in higher-level languages.*

Programming Language Implementation

- A language implementation system requires a large collection of programs, called the operating systems, which supplies high-level primitive than those of the machine language
  - System resource management
  - Input and output operations
  - A file management system
  - Text and/or program editors
  - A variety of other commonly needed functions

- A language implementation system interfaces with the operating system rather than directly with the processor.

Programming Language Implementation

- A programming language implementation is a system for executing the programs.

- Three general approaches to implementation:
  - ***Compilation***
    - A compiler transforms a program written in a particular programming language and turns them into machine language
  - ***Interpretation***
    - An interpreter executes instructions written in a particular programming language
  - ***Hybrid***