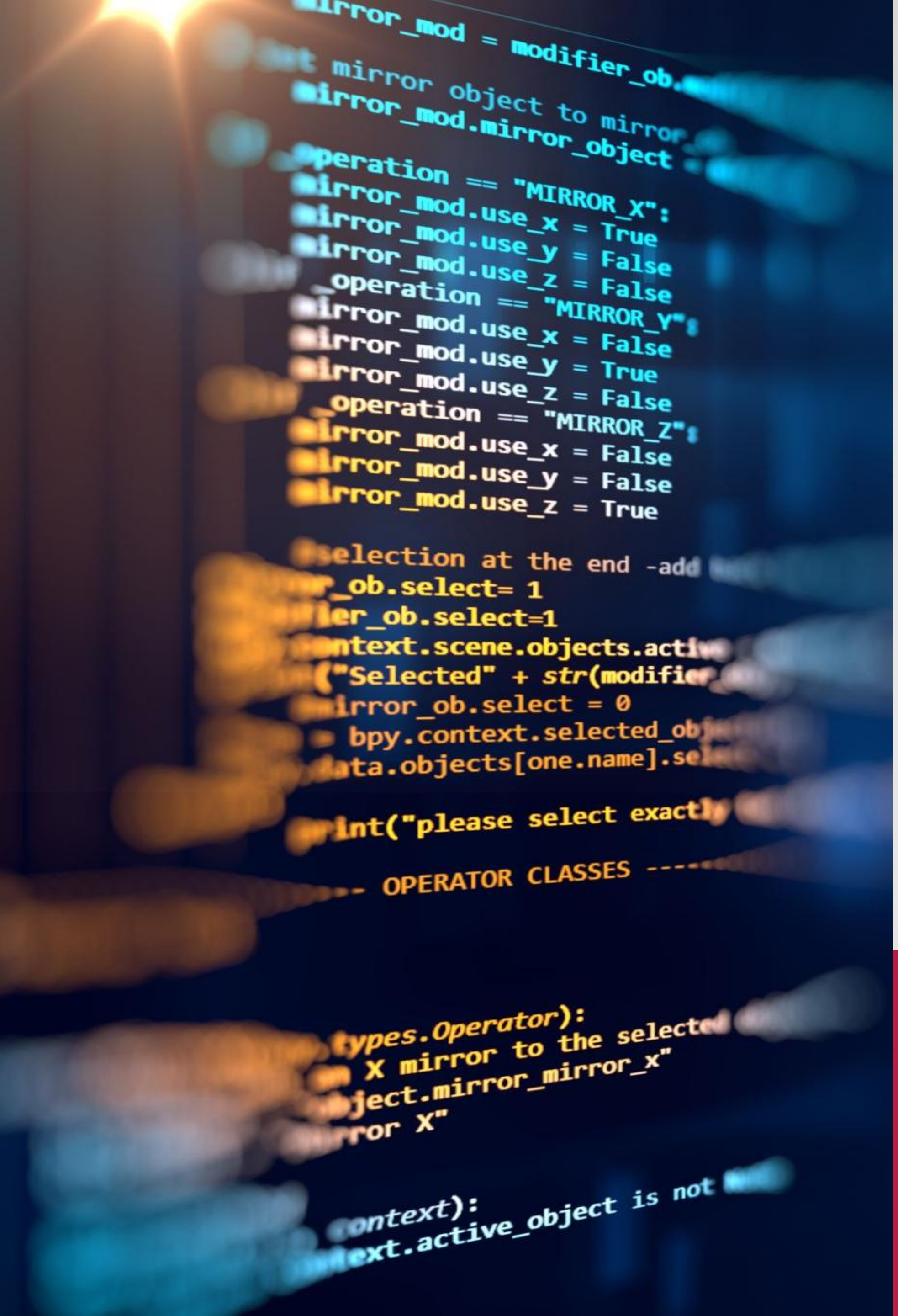


COMP 348 PRINCIPLES OF PROGRAMMING LANGUAGES

LECTURE 0 – INTRODUCTION

Dept of Computer Science and Software Engineering, Concordia University

Fall 2020

Introduction to Programming Languages

Overview

added

What is a Programing Language?

- A **“user-interface”** to a computer
- It provides a linguistic framework for describing computations
- It is a **notational system** for describing computations in human-readable form.

- A Programming Language is a set of rules that provides a way of telling a computer what operations to perform.

- “A programming language is a notational system for describing computation in a machine-readable and human-readable form”

  -- Kenneth Louden

added

What is a Programing Language?

- Similar to English Language, a programming language has its own **words** and **grammar**. The grammatical rules are called syntax. Each programming language has a different set of rules and syntax.
- What are there many different programming languages?
  - Different programming languages are designed for different types of programs.

- Concepts related to Programming Languages
  - Paradigms
  - Semantics
  - Foundations
  - Implementation

added

Generations of Programming Languages

- 1GL: Machine Code 0s and 1s
  - Intel x86, x64 Instruction Set
- 2GL: Assembly Language (one to one mapping to machine code)
  - Intel x86, x64 Assembly Language (MASM)
- 3GL: High-Level Machine-Independent Languages
  - FORTRAN, PASCAL, PL/I, C, …
- 4GL: Domain specific application generators
  - Oracle 4GL, …
- 5GL: varies

“**A fifth-generation programming language (5GL)** is any programming language based on problem-solving using **constraints** given to the program, rather than using an algorithm written by a programmer”. Most constraint-based and logic programming languages and some other declarative languages are fifth-generation languages

added

How do Programming Languages Differ?

- Common Constructs
  - basic data types (numbers, etc.); variables; expressions; statements; keywords; control constructs; procedures; comments; errors ...

- Uncommon Constructs:
  - type declarations; special types (strings, arrays, matrices, ...); sequential execution; concurrency constructs; packages/modules; objects; general functions; generics; modifiable state; ...

added

Programming Paradigms

- Common programming paradigms:

Imperative style:	program = algorithms + data *good for decomposition*
Functional style:	program = functions $\circ$ functions *good for reasoning*
Logic programming style:	program = facts + rules *good for searching*
Object-oriented style:	program = objects + messages *good for modeling(!)*

added

Programming Paradigms

Examples:

- Imperative Programming (C)
- Object Oriented Programming (C++)
- Logic/Declarative Programming (Prolog)
- Functional Programming (LISP)

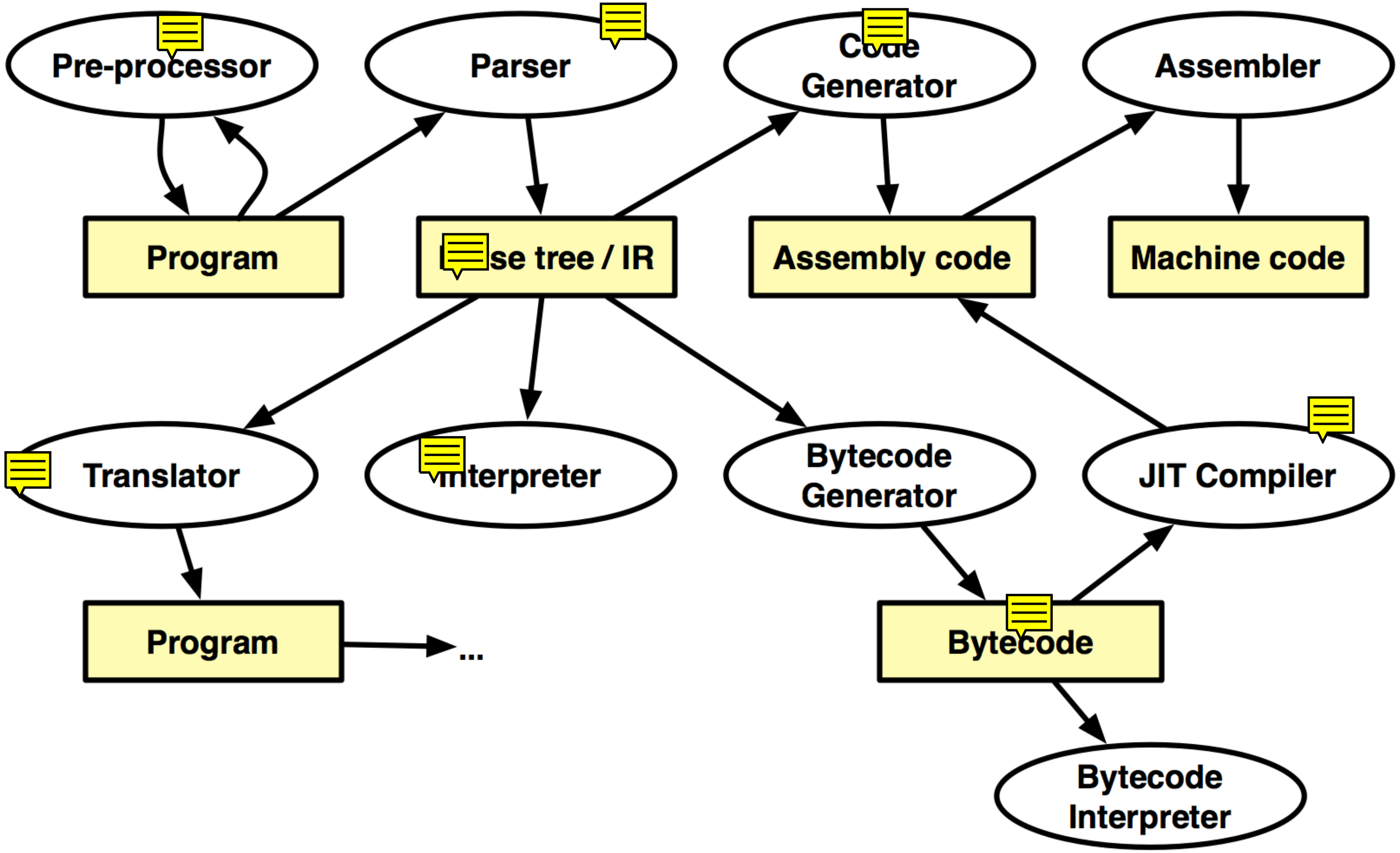
added

Some Other Categories of Languages

- Scripting Languages
  - JavaScript, PHP, ASP
- Command Languages
  - SH, CSH, BASH
- Text Processing Languages
  - LaTex, PostScript
- Markup Languages
  - HTML, XML
- Query Languages
  - SQL
- Visual languages
- Modeling languages
  - UML
- Dynamic Languages
  - Computation of code in runtime, common methods: eval/reflection/marcos/.
- Domain-specific Languages

added

Compilers and Interpreters



added

A Brief Chronology

First programming languages

- 1951 – Regional Assembly Language
- 1952 – Autocode
- 1954 – IPL (forerunner to LISP)
- 1955 – FLOW-MATIC (led to COBOL)
- 1957 – **FORTRAN** (first compiler)
- 1957 – COMTRAN (precursor to COBOL)
- 1958 – **LISP**
- 1958 – ALGOL 58
- 1959 – FACT (forerunner to COBOL)
- 1959 – **COBOL**
- 1959 – RPG
- 1962 – APL
- 1962 – Simula
- 1962 – SNOBOL
- 1963 – CPL (forerunner to C)
- 1964 – Speakeasy
- 1964 – **BASIC**
- 1964 – **PL/I**
- 1966 – JOSS
- 1966 - MUMPS

Establishing fundamental paradigms

- 1967 – **BCPL (forerunner to C)**
- 1968 – Logo
- 1969 – **B (forerunner to C)**
- 1970 – **Pascal**
- 1970 – Forth
- 1972 – **C**
- 1972 – Smalltalk
- 1972 – Prolog
- 1973 – ML
- 1975 – Scheme
- 1978 – **SQL** (a query language, later extended)

added

A Brief Chronology

1980s: consolidation, modules, performance

- 1980 – C++ (as **C with classes**, renamed in 1983)
- 1983 – Ada
- 1984 – **Common Lisp**
- 1984 – **MATLAB**
- 1984 – **dBase III**, dBase III Plus (Clipper and FoxPro as FoxBASE, later developing into Visual FoxPro)
- 1985 – Eiffel
- 1986 – **Objective-C**
- 1986 – LabVIEW (Visual Programming Language)
- 1986 – Erlang
- 1987 – **Perl**
- 1988 – Tcl
- 1988 – Wolfram Language (as part of Mathematica, only got a separate name in June 2013)
- 1989 – FL (Backus)

1990s: The Internet Age

- 1990 – Haskell
- 1991 – **Python**
- 1991 – **Visual Basic**
- 1993 – Lua
- 1993 – **R**
- 1994 – CLOS (part of ANSI Common Lisp)
- 1995 – **Ruby**
- 1995 – Ada 95
- 1995 – **Java**
- 1995 – Delphi (Object Pascal)
- 1995 – **JavaScript**
- 1995 – **PHP**
- 1997 – Rebol

added

A Brief Chronology

Current Trends:

- 2000 – ActionScript
- 2001 – **C#**
- 2001 – D
- 2002 – Scratch
- 2003 – **Groovy**
- 2003 – Scala
- 2005 – F#
- 2006 – PowerShell
- 2007 – Clojure

- 2009 – **Go**
- 2010 – Rust
- 2011 – Dart
- 2011 – **Kotlin**
- 2011 – Elixir
- 2012 – Julia
- 2012 - **TypeScript**
- 2014 – **Swift**

More info may be found here:

https://en.wikipedia.org/wiki/History_of_programming_languages

added

Examples of Programming Languages

- **The “Hello World”**

The tradition of using the phrase "Hello, World!" as a test message was influenced by an example program in the seminal 1978 book The C Programming Language.

```
main()
{
    printf("hello, world\n");
}
```

"hello, world", and was inherited from a 1974 Bell Laboratories internal memorandum by Brian Kernighan, Programming in C: A Tutorial

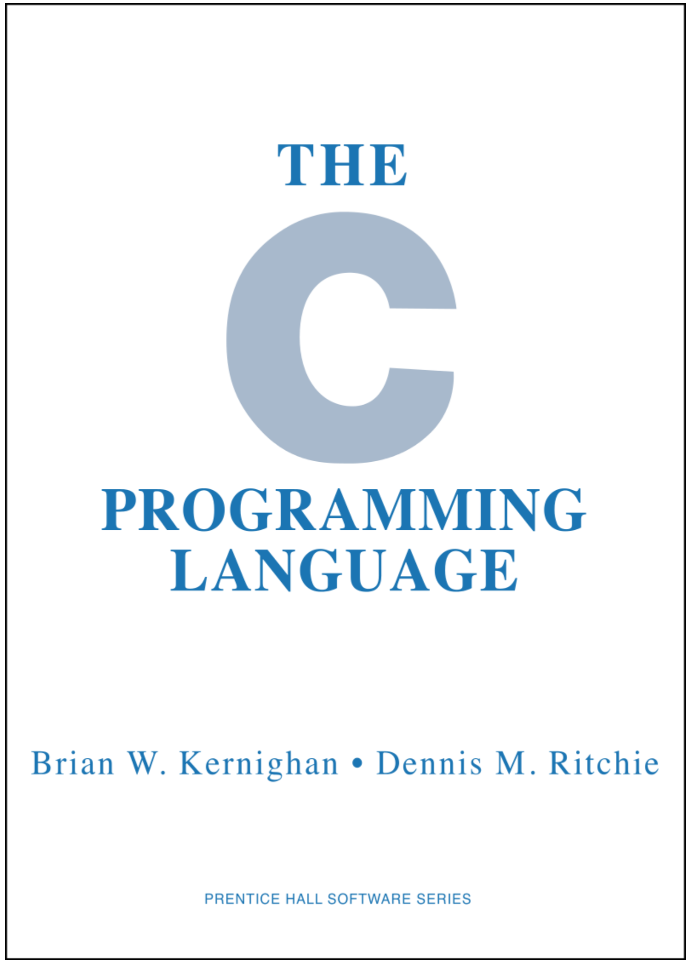
added

The C Programming Language

```
main()
{
    printf("hello, world\n");
}
```

"Hello, World!" program by Brian Kernighan (1978)

Available @
https://archive.org/details/TheCProgrammingLanguageFirstEdition



THE C PROGRAMMING LANGUAGE

Brian W. Kernighan • Dennis M. Ritchie

PRENTICE HALL SOFTWARE SERIES

Source wikipedia

added

Examples of Programming Languages

- **FORTRAN**

```fortran
program helloworld
    print *, "Hello, World!"
end program helloworld
```

- **BASIC**

```basic
10 PRINT "Hello World!"
20 GOTO 10
```

- **PROLOG**

```prolog
hello :- printstring("HELLO WORLD!!!!").

printstring([]).
printstring([H|T]) :- put(H), printstring(T).
```

added

Examples of Programming Languages

- **LISP**

```
(DEFUN HELLO-WORLD ()
   (PRINT (LIST 'HELLO 'WORLD)))
```

- **KOTLIN**

```
// Hello, World! Example
fun main() {
  val scope = "World"
  println("Hello, $scope!")
}
```

- **PL/SQL**

```
SELECT 'HELLO WORLD' FROM DUAL
```

added

Acknowledgement

- https://en.wikipedia.org/wiki/History_of_programming_languages
- https://en.wikipedia.org/wiki/Programming_paradigm
- https://www.slideshare.net/VarunGarg7/lect-1-introduction-to-programming-languages
- http://www.cs.cmu.edu/~violetta/CMP131/Lectures/Week1.3.ppt