

CSC 4101: Programming Languages

Gerald Baumgartner
gb@csc.lsu.edu

Introduction

Textbook, Chapter 1

2

Why Are We Interested in Programming Languages?

Programming in machine code is way too tedious/error-prone

3

Why So Many Languages?

- Evolution
 - From goto to loops, case statements
- Personal Preference
 - Syntax
 - Loops vs. recursion
 - Pointers vs. recursive data types
- Special Purposes

7

Application Domains

- Scientific applications (Fortran, TCE)
- Business applications (Cobol)
- Artificial intelligence (Lisp)
- Systems programming (C, C++)
- Web service programming (Java, C#)
- Very High-Level Languages (perl)
- Special purpose languages (make, sh)

8

What Makes a Language Succeed?

- Expressive Power
- Ease of Use for Novice
- Ease of Implementation
- Open Source
- Availability of Compilers, Libraries
- Economics, Patronage, Inertia
- Syntax that looks like C

9

Language Design Issues

- Readability (*p++)
- Abstractions (functions, classes)
- Orthogonality (no special cases)
- Reliability (type checking)
- Cost (training programmers)

10

Programming Paradigms

- Imperative (C, Pascal, etc.)
- Functional (Lisp, ML, Haskell)
- Logic (Prolog)
- Object-Oriented (C++, Java, CLOS)

11

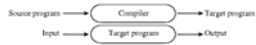
Why Do We Study Programming Languages?

- Understand obscure language features
- Choose among ways to express ideas
- Make good use of debuggers, other tools
- Simulate nice features in other languages
- Choose appropriate language for problem
- Learn new languages faster
- Design simple languages

12

Implementation Methods

- **Compilation (C, ML)**



- **Interpretation (early Lisp)**

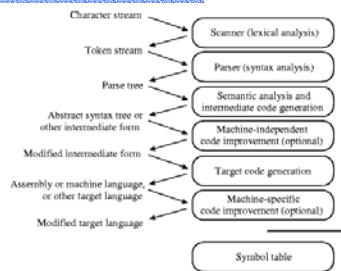


- **Hybrid Systems (early Java)**



13

Overview of Compilation



14

Source Code for GCD

```
program gcd(input, output);  
var i, j: integer;  
begin  
  read(i, j);  
  while i <> j do  
    if i > j then i := i - j  
    else j := j - 1  
  writeln(j);  
end.
```

15
