

# Design and Selection of Programming Languages

SFWR ENG 3E03, Fall 2005

WOLFRAM KAHL

*Department of Computing and Software  
Faculty of Engineering  
McMaster University*

## Are Programming Languages Important in SE?

*Programming happens only in the last project phases — how can the programming language make any difference?*

- Coding starts from design specifications
- **Coding considerations can influence design decisions**
- Code has to be verified against design specifications
- Code needs to be accessible to testing against specifications, requirements ...
- Code needs to be accessible to **maintenance**
- Code fragments should be accessible to **reuse**

## Why Study Programming Languages?

- Understand the **purpose** of programming languages
- Appreciate the advantages and disadvantages of various languages
- Understand the need for and use of **typical programming constructs**
  - this makes learning a new language much easier
- Be able to **choose** an appropriate language for a given task
- Be able to *cope with an inappropriate language* when forced to use one
- Appreciate that implementation aspects need to be separated from language aspects.

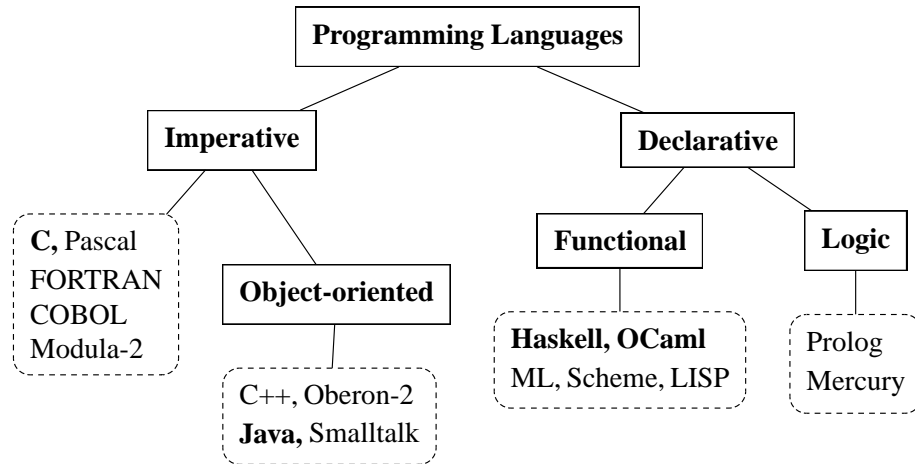
## What Is a Programming Language?

- Syntax** — What a program looks like
- Semantics** — What a program means
- Pragmatics* — How people use the language
- Implementation* — How a program executes

## What Kinds of Programming Languages are There?

**Imperative** — “telling the machine what to do”

**Declarative** — “telling the machine what to achieve”



## Programming Language Paradigms

### Imperative Programming Languages

Statement oriented languages

Every statement changes the machine state

### Functional (Applicative) Programming Languages

Goal is to understand the function that produces the answer

Function composition is major operation

Programming consists of building the function that computes the answer

### Rule-Based (Logical) Programming Languages

Specify rule that specifies problem solution (Prolog, BNF Parsing)

Other examples: Decision procedures, Grammar rules (BNF)

Programming consists of specifying the attributes of the answer

### Object-oriented languages

Organising the state into *objects* with individual state and behaviour

Message passing paradigm (instead of subprogram call)

## Historical Development of Programming Languages

Emphasis has changed:

– from making life easier for the computer

– to making it easier for the programmer.

**Easier for the programmer** means:

– Use languages that facilitate writing **error-free programs**

– Use languages that facilitate writing programs that are **easy to maintain**

**Goal** of language development:

– Developers concentrate on design (or even just specification)

– Programming is trivial or handled by computer

(*executable specification languages, rapid prototyping*)

## Grading

End of September?	<b>Midterm 1</b>	20% or 10%
Mid October	<b>Midterm 2</b>	20% or 10%
November	<b>Midterm 3</b>	20% or 10%
December	<b>Final</b>	(between 40% and 70%)

“X% or Y%” means: If midterm result is better than the result of the final, then the midterm counts X%, otherwise Y%.