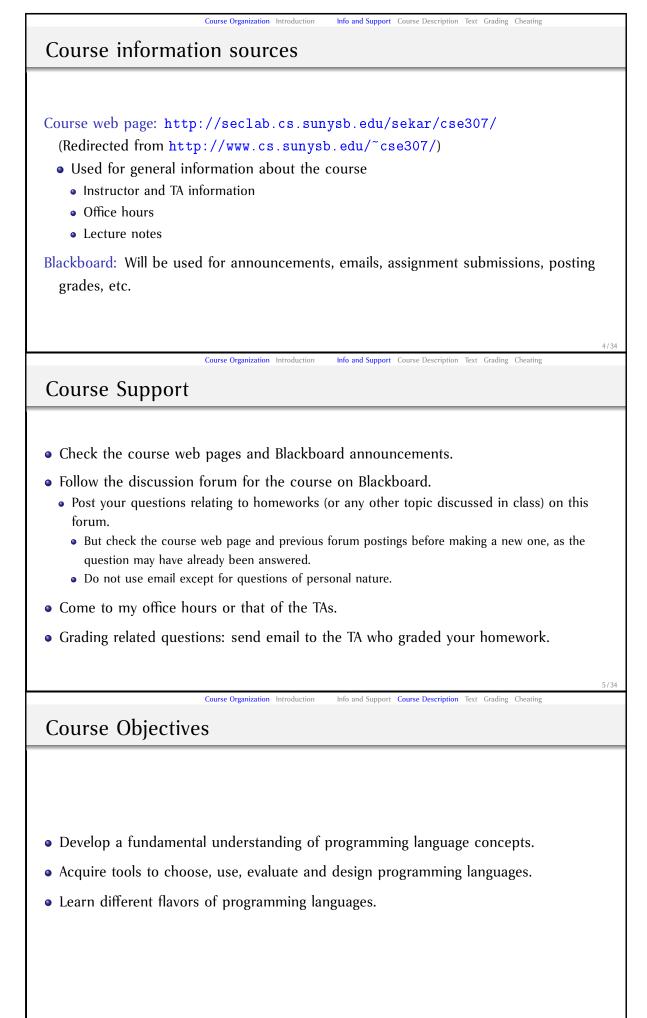
Course Organization	Introduction	
	f D	
CSE 307: Principie	s of Programming Languages	
	Spring 2015	
	R. Sekar	
Course Organization	Introduction	1/34
Topics		
Торісэ		
	Cheating	
1. Course Organization	2. Introduction	
Info and Support Course Description	Languages and Characteristics Paradigms	
Text	History	
Grading	Design Criteria	
Course Operation	biological control Course Description Test Carlier Charter	2/34
Course Organization	Introduction Info and Support Course Description Text Grading Cheating	
	Section 1	
	Section	
Cour	se Organization	
		2/24



Course Organization Introduction Info and Support Course Description Text Grading Cheating	
What will you learn in CSE 307?	
 Programming Language Concepts Values, Binding, Scopes, Naming, Programming Paradigms Object-oriented, Functional, Logic Runtime environments, Interpreters and Compilers 	
Course Organization Introduction Info and Support Course Description Text Grading Cheating	7/34
Programming Languages Covered	
Imperative: C, [Pascal] Object-oriented: C++, Javascript, Java Functional: OCAML, Use of functional style in Python, [Haskell] Logic: Prolog	8/34
Course Organization Introduction Info and Support Course Description Text Grading Cheating Textbooks	
 Required: Kenneth C. Louden, <i>Programing Languages Principles and Practice</i>, Second Edition, Thomson Publishers. You can buy the 3rd edition if you cannot find the 2nd edition 	

• But I find used versions of the second edition on Amazon for \$1 or so!

How the course is run

- Approximately one homework every two weeks
 - Non-programming homeworks about programming language concepts.
 - Short programming assignments to learn programming in new languages.
 - Larger (around 500 lines) programming assignments: writing interpreters to solidify the understanding programming language concepts.
- You can skip or drop one assignment in the whole semester
- Grading (approximate)
 - 70% exams: Two midterms (15% to 18% each) plus Final (approx. 35%)
 - 30% homeworks, assignments, quizzes and class participation.

Course Organization Introduction

• To receive a good grade, you must do well *individually* in each component

Academic Integrity

• Do not copy from any one, or any source (on the Internet or elsewhere)

Course Organization Introduction

• The penalty for cheating is an F-grade, plus referral to graduate school. *No exception,* regardless of the "amount" of copying involved.

Info and Support Course Description Text Grading Cheating

Languages and Characteristics Paradigms History Design Criteria

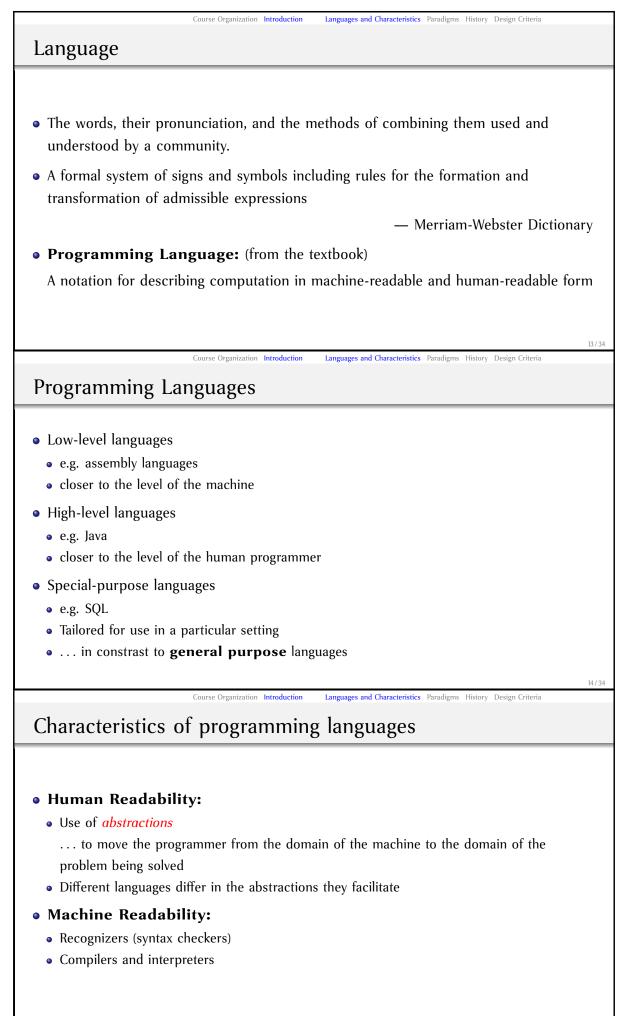
- In addition, if you cheat, you will be unprepared for the exams, and will do poorly.
- To encourage you to work on your own, we scale up assignment scores by about 10% to 25%

11/34

10/34

Section 2

Introduction



Data Abstraction

- Basic Data Abstraction:
 - Common, "atomic" data values such as integers;
 - places to store such values (e.g. "variables");
 - notation to indicate the association between places, their names and values.
- Structured Data Abstraction:
 - Group or collection of related data values
 - e.g. arrays, records, etc.
- Unit Data Abstraction:
 - Encapsulating related data values and structures into induvidual program units

Course Organization Introduction

• e.g. modules and packages.

Control Abstraction

- Basic Control Abstraction:
 - Most fundamental of control (e.g. data movement)
 - Statements (e.g. x = x + 1)
- Structured Control Abstraction:
 - Combine basic controls into more powerful groups
 - Control *structures* such as if, while, case/switch etc.
 - Abstraction of a group (sequence) of actions into a single action (e.g. procedures and methods)
- Unit Control Abstraction:
 - Same as in data abstraction: packages and modules

Course Organization Introduction

17/34

16/34

The study of languages

Language Definition

- Syntax (structure)
- Semantics (meaning)

Language Features

- Control structures
- Data structures
- Extensions

Language Processing

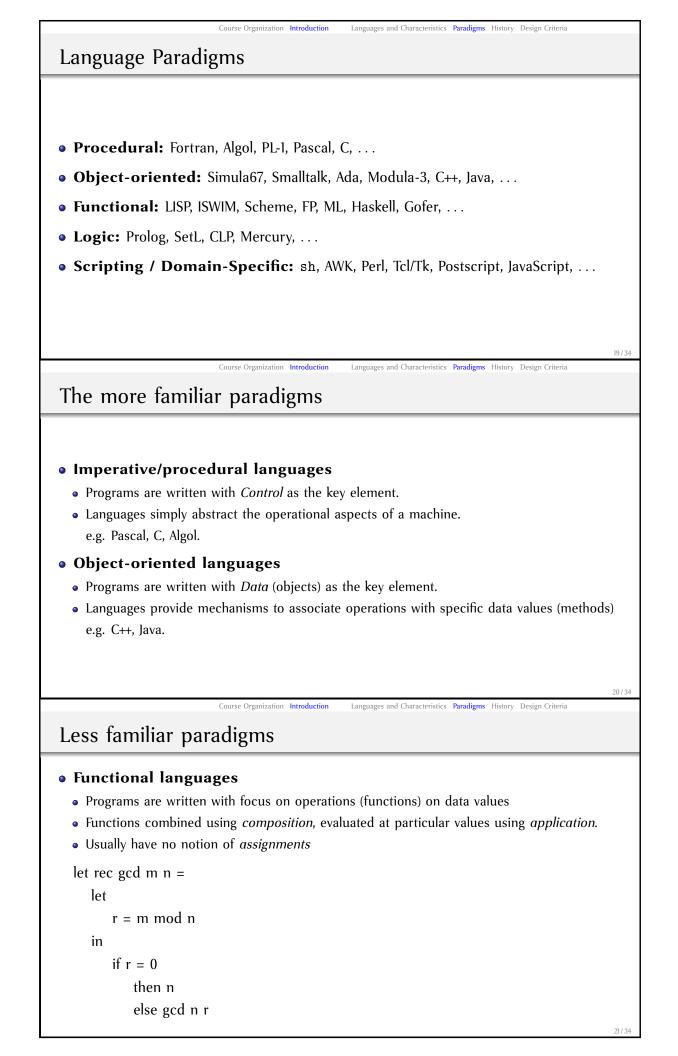
Languages and Characteristics Paradigms History Design Criteria

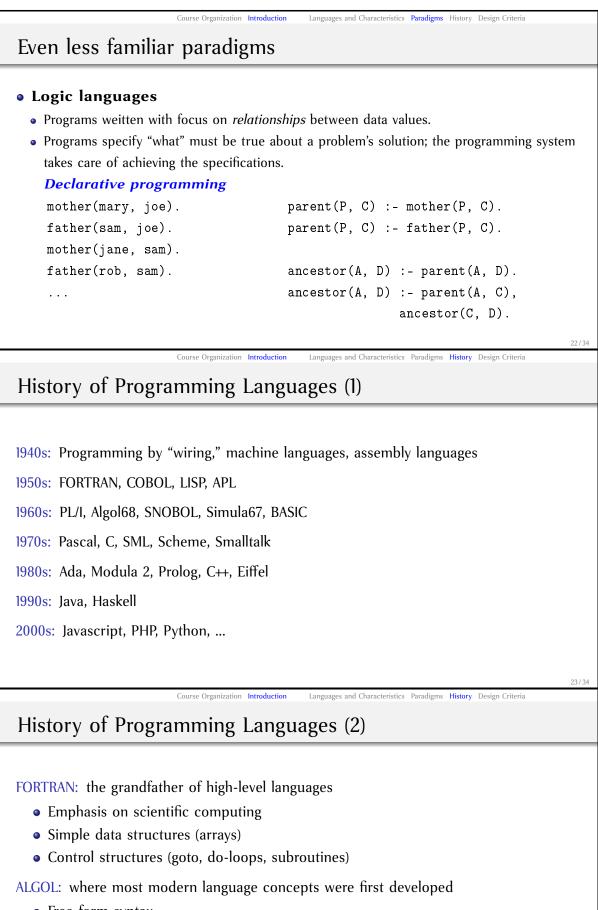
Languages and Characteristics Paradigms History Design Criteria

- Translation
- Interpretation
- Runtime environment

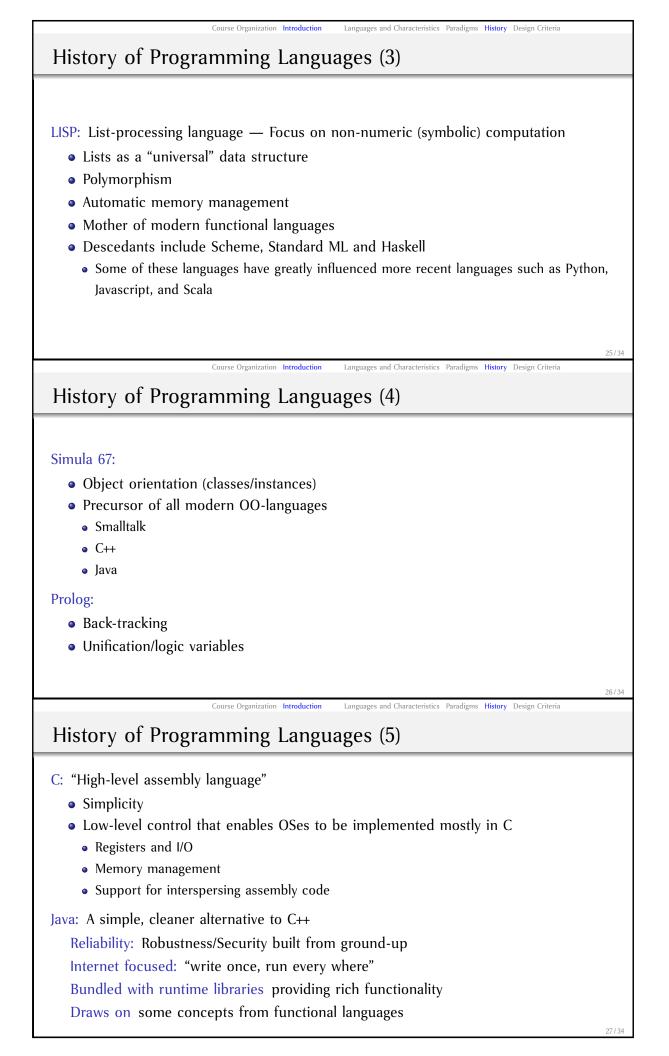
• Language Design

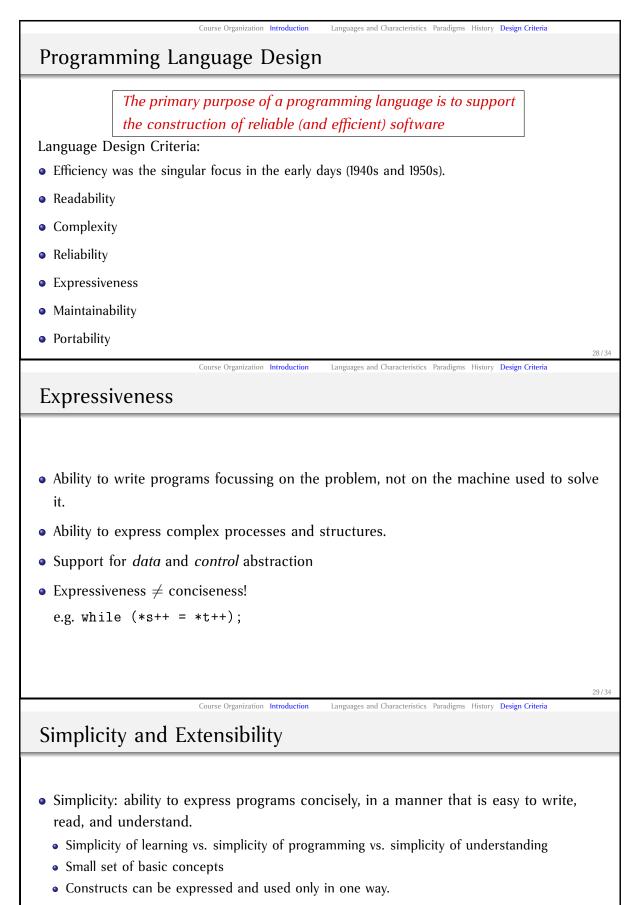
- Understandability
- Simplicity and Expressiveness
- Efficiency
- Portability
- Security/Error-checking





- Free-form syntax
- Block-structure
- Type declarations
- Recursion





- Extensibility: ability to add new features to the language
 - Data type definition in Pascal, Modula, Ada, ...
 - Definition of new operators (or reuse existing operators such as '+') in SML, Prolog, Haskell, . . .

Course Organization Introduction Languages and Characteristics Paradigms History Design Criteria	
Regularity	
negularity	
 Generality: Operations/constructs available for all closely related cases: in C: Compare two integers with == but not two structures/arrays in Java: Can make collections of objects (e.g. Integer) but not primitive values (e.g. int) 	
 Orthogonality: Constructs can be combined in a meaningful way: 	
 in C: All parameters are passed by value <i>except arrays</i> 	
• in Java: A class can have static members but an abstract class cannot.	
• Uniformity: Constructs appear and behave consistently:	
• in C, Java: = means "assignment" while == is a comparison.	
 "Law of least astonishment" 	
	31/34
Course Organization Introduction Languages and Characteristics Paradigms History Design Criteria	
Efficiency	
 Efficiency of executable code Efficiency of translation Efficiency of programming Reusabilty Reliability Security Maintainability 	32/34
Consistency, Precision and Security	
 Use of accepted notations and notions D0 9 I = 1, 10 D0 9 I = 1. 10 Availability of well-specified standards When is int same as long? short? Are structures byte aligned? or word aligned? Constructs to build programs that cannot be subverted Array bounds checks Safety in types 	

Error Detection and Correction

- Catch programming errors at compile-time
 - Strong type system
 - Memory safety
- Constructs to handle usage errors
 - Exception handling mechanism
- Mechanisms to test and uncover errors
 - Reflection ...

34/34