CS2303 C14 Systems Programming Concepts





Introduction





Introduction

- Survey and TA/SA Introductions
- Pause to Look Backwards and Forwards
- . Course Objectives
- Course Operation/Expectations
- Course Plan and Syllabus
- . Systems Concepts
- Higher Level Language History
- · 'Old' Development Environment
 - C and C++

Quick Look Backwards/Forwards

- . Computing Devices
 - From mainframes to PCs to smart phones to ??
- . Changes in WPI CS Curriculum
- . Instructor
- . Students
 - Expected Background
 - Going Forward
 - Your Future





CS2303 Course Objectives

- To expose students to the low level systems interface 'grunge' clearly visible in C.
- To learn to program in C++ by learning to program in C first.
- To further develop the ability to design programs with emphasis on the abstract view of data structures.
- To get experience with low level implementation of data structures in C.



CS2303 Course Objectives

- To learn the advantages of programming in an object-oriented language such as C++.
- . To experience programming in the





CS2303 Course Objectives

Pointers!!



Course Operation/Expectation

- The course web page is an essential student asset.
- * Students are responsible for all information on web page!
- .5 Required Labs
- .5 Programming Assignments
- .2 Closed Book Exams

Course Plan and Syllabus

- . To cover the details of C briskly.
 - Assumes students already have an understanding of iteration and conditional constructs.
 - Using only C I/O {grunge as promised!} at first.
- To introduce data structures in C by doing at least one program with structs and call by value.
- . To finish up with as much C++ as possible.
- {Note reading of the textbook will require jumping around during the C portion of the course.}



Systems Concepts

- The goal of this programming course is to expose the students to places where the software and hardware meet or where the application interfaces with the operating system (OS).
- A 'systems viewpoint' includes resource management (CPU and memory), process scheduling, concurrency and performance.
- {But this is too much material for this instance of the course!}



Systems Performance Viewpoint

- The assignments include simulation and introduce two system performance concerns - efficiency and fairness.
- The other important approach to appreciate is the computer scientist abstraction concept of insulating interfaces from 'under-the-hood' details (e.g., virtual memory and loaders).



1.5 Types of Programming Languages

- Although assembly-language code is clearer to humans, it's incomprehensible to computers until translated to machine language.
- To speed the programming process even further, high-level languages were developed in which single statements could be written to accomplish substantial tasks.
- High-level languages allow you to write instructions that look almost like everyday English and contain commonly used mathematical expressions.
- Translator programs called <u>compilers</u> convert high-level language programs into machine language.
- **Interpreter** programs were developed to execute high-level language programs directly, although more slowly than compiled programs.
- Scripting languages such as JavaScript and PHP are processed by interpreters.

Copyright © Pearson, Inc. 2013. All Rights Reserved.



Higher Level Programming Languages History

Fortran _w	1957		COBOL	1959
Algol	1960 1	968	Lisp	1959
PL1	1964		APL	1962
Pascalw	1970		SNOBOL	1967
C _w	1972		Prolog	1972
Basic	1975		Schemew	1975
C++ _w	1986		ADA	1983
Javaw	1995		Python _w	1989



C Program Development Environment

Standard Steps

- 1. Edit
- 2. Preprocess
- 3. Compile
- 4. Link
- 5. Load
- 6. Execute

Deitel & Deitel

© 2007 Pearson Ed -All rights reserved.



Fig. 1.1 | Typical C development environment.



Systems Programming

Introduction

User Memory Protection



Virtual Memory



Review of Introduction

- . Course Objectives
- Course Operation/Expectations
- . Course Plan and Syllabus
- . C, data structures, C++
- . Systems Viewpoint {more later}
- Program Development Environment

