CS-101(New): Principles of Programming Languages
[Total Lectures: 48 Hours]

Course Prerequisites:

It is assumed that student learning this course have the following background:

- Experience with an OOP language (such as Java or C++)
- Experience with a procedural language (such as C)
- Working knowledge of C, C++, and Java programming.
- Basic algorithms and data structure concepts.

Why to study this course?

- To allow Informed Design Decisions
- Gives insight when debugging
- Permits effective use of compilers/linkers interpreters and language oriented tools.
- Helps to understand how language features work.
- Learn features, emulate missing features.
- Develop a greater understanding of the issues involved in programming language design and implementation
- Develop an in-depth understanding of functional, logic, and object-oriented programming paradigms
- Implement several programs in languages other than the one emphasized in the core curriculum (Java/C++)
- Understand design/implementation issues involved with variable allocation and binding, control flow, types, subroutines, parameter passing
- Develop thorough understanding of the compilation process
- To introduce several different paradigms of programming
- To gain experience with these paradigms by using example programming languages
- To understand concepts of syntax, translation, abstraction, and implementation

Course Objectives:

- This course will prepare you to think about programming languages analytically:
  - Separate syntax from semantics
  - Compare programming language designs
  - Learn new languages more quickly
  - Use standard vocabulary when discussing languages
  - Understand basic language implementation techniques
- This course focuses on both:
  - Theory is covered by the textbook readings, lectures, and on the tests
  - Implementation is covered by the homework assignments

Unit 1. Introduction [T1 chap. 1] [2]

- The Art of Language Design [T1 1.1]
- The Programming Language Spectrum [T1 1.2]
- Why Study Programming Languages? [T1 1.3]
- Compilation and Interpretation [T1 1.4]
- Programming Environments [T1 1.5]

Unit 2. Non-Imperative Programming Models: Functional, Logic Languages

[Text books 3, 4] **[10]**

Common LISP

- Basic LISP Primitives (FIRST, REST, SETF, CONS, APPEND, LIST, NTHCDR, BUTLAST,LAST, LENGTH, REVERSE, ASSOC)
- Procedure definition and binding, DEFUN, LET
- Predicates and Conditional,
  EQUAL, EQ, EQL, =, MEMBER, LISTP, ATOM, NUMBERP,
  SYMBOLP, NIL, NULL, IF, WHEN, UNLESS,
  COND, CASE
- Procedure Abstraction and Recursion

[**T 4**]

Turbo Prolog

Introduction, facts, Objects and Predicates, Variables, Using Rules, Controlling execution fail and cut predicates

[**T3** chapter 1 through 9 except chapter 2]

Unit 3. Names, Scopes, and Bindings [T1 chap.3] [5]

The Notion of Binding Time [T1 chap.3.1]

Object Lifetime and Storage Management : [T1 chap. 3.2]

Static Allocation, Stack-Based Allocation, Heap-Based Allocation, Garbage Collection

Scope Rules T1 chap. 3.3]

Static Scoping, Nested Subroutines, Declaration Order, Dynamic Scoping

The meaning of Names in a Scope [T1 chap. 3.5]

Aliases, Overloading, Polymorphism and Related Concepts

The Binding of Referencing Environments [T1 chap. 3.6]

Subroutine Closures, First-Class Values and Unlimited Extent, Object Closures

Macro Expansion [T1 chap. 3.7]

Unit 4. Control Flow [T1 chap.6] [5]

Expression Evaluation [T1 6.1]

Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation

Structured and Unstructured Flow [T1 6.2]

Structured Alternatives to goto

Sequencing [T1 6.3]

Selection [T1 6.4]

Short-Circuited Conditions, Case/Switch Statements

Iteration [T1 6.5]

Enumeration-Controlled Loops, Combination Loops, Iterators, Logically Controlled Loops

Recursion [T1 6.6]

Iteration and Recursion, Applicative- and Normal-Order Evaluation

Unit 5. Data Types [T2 chap.6] **[8]**

Introduction T2 6.1]
Primitive Data Types [T2 6.2]
- Numeric Types [T2 6.2.1]
- Integer [T2 6.2.1.1]
- Floating point [T2 6.2.1.2]
- Complex [T2 6.2.1.3]
- Decimal [T2 6.2.1.4]
- Boolean Types [T2 6.2.2]
- Character Types [T2 6.2.3]
- Character String Types [T2 6.3]
  - Design Issues [T2 6.3.1]
  - Strings and Their Operations [T2 6.3.2]
  - String Length Operations [T2 6.3.3]
  - Evaluation [T2 6.3.4]
  - Implementation of Character String Types [T2 6.3.5]

User defined Ordinal types [T2 6.4]
- Enumeration types [T2 6.4.1]
  - Designs
  - Evaluation
- Subrange types [T2 6.4.2]
  - Ada's design
  - Evaluation
- Implementation fo used defined ordinal types [T2 6.4.3]

Array types [T2 6.5]
- Design issues [T2 6.5.1]
- Arrays and indices [T2 6.5.2]
- Subscript bindings and array categories [T2 6.5.3]
- Heterogeneous arrays [T2 6.5.4]
- Array initialization [T2 6.5.5]
- Array operations [T2 6.5.6]
- Rectangular and Jagged arrays [T2 6.5.7]
- Slices [T2 6.5.8]
- Evaluation [T2 6.5.9]
- Implementation of Array Types [T2 6.5.10]

Associative Arrays [T2 6.6]
- Structure and operations [T2 6.6.1]
- Implementing associative arrays [T2 6.6.2]

Record types [T2 6.7]
- Definitions of records [T2 6.7.1]
- References to record fields [T2 6.7.2]
- Operations on records [T2 6.7.3]
- Evaluation [T2 6.7.4]
- Implementation of Record types [T2 6.7.5]

Union Types [T2 6.8]
- Design issues [T2 6.8.1]

Discriminated versus Free unions [T2 6.8.2]
Evaluation [T2 6.8.4]
Implementation of Union types [T2 6.8.5]
Pointer and Reference Types [T2 6.9]
Design issues [T2 6.9.1]
Pointer operations [T2 6.9.2]
Pointer problems [T2 6.9.3]
Dangling pointers
Lost heap dynamic variables
Pointers in C and C++ [T2 6.9.5]
Reference types [T2 6.9.6]
Evaluation [T2 6.9.7]
Implementation of pointer and reference types [T2 6.9.8]
Representation of pointers and references
Solution to dangling pointer problem
Heap management

Unit 6. Subroutines and Control Abstraction [T2 chap.9,10] [5]

Fundamentals of Subprograms [T2 9.2 (excluding 9.2.4)]
1Design Issues for subprograms [T2 9.3]
Local Referencing Environments [T2 9.4]
Parameter-Passing Methods [T2 9.5]
Parameters That are Subprograms [T2 9.6]
Overloaded Subprograms [T2 9.7]
Generic Subroutines [T2 9.8]
Generic Functions in C++ [T2 9.8.2]
Generic Methods in Java [T2 9.8.3]
Design Issues for Functions [T2 9.9]
User-Defined Overloaded Operators [T2 9.10]
Coroutines [T2 9.10]
The General Semantics of Calls and Returns [T2 10.1]
Implementing "Simple" Subprograms [T2 10.2]
Implementing Subprograms with Stack-Dynamic Local Variables [T2 10.3]
Nested Subprograms [T2 10.4]
Blocks [T2 10.5]
Implementing Dynamic Scoping [T2 10.6]

Unit 7. Data Abstraction and Object Orientation [T1 chap.9] [8]

Object-Oriented Programming [T1 9.1]
Encapsulation and Inheritance [T1 9.2]
Modules, Classes, Nesting (Inner Classes), Type Extensions, Extending without Inheritance
Initialization and Finalization [T1 9.3]
Choosing a Constructor, References and Values, Execution Order, Garbage Collection
Dynamic Method Binding [T1 9.4]
Virtual- and Non-Virtual Methods, Abstract Classes, Member Lookup,

Polymorphism, Object Closures

Multiple Inheritance [T1 9.5]

Semantic Ambiguities, Replicated Inheritance, Shared Inheritance, Mix-In Inheritance

Unit 8. Concurrency [T2 chap. 13] [5]

Introduction

Multiprocessor Architecture [T2 13.1.1]
Categories of concurrency [T2 13.1.2]
Motivations for studying concurrency [T2 13.1.3]

Introduction to Subprogram-level concurrency

Fundamental concepts [T2 13.2.1]
Language Design for concurrency. [T2 13.2.2]
Design Issues [T2 13.2.3]

Semaphores

Introduction [T2 13.3.1]
Cooperation synchronization [T2 13.3.2]
Competition Synchronization [T2 13.3.3]
Evaluation [T2 13.3.4]

Monitors

Introduction [T2 13.4.1]
Cooperation synchronization [T2 13.4.2]
Competition Synchronization [T2 13.4.3]
Evaluation [T2 13.4.4]

Message Passing

Introduction [T2 13.5.1]
The concept of Synchronous Message Passing [T2 13.5.2]

Java Threads

The **Thread** class [T2 13.7.1]
Priorities [T2 13.7.2]
Competition Synchronization [T2 13.7.3]
Cooperation Synchronization [T2 13.7.4]
Evaluation [T2 13.4.5]

Text Books:

T1. Scott Programming Language Pragmatics, 3e(With CD) ISBN 9788131222560 Kaufmann Publishers, An Imprint of Elsevier, USA
T2. Concepts of Programming Languages, Eighth Edition by Robert W. Sebesta, Pearson Education.
T3. Introduction to Turbo Prolog by Carl Townsend
T4. LISP 3rd edition by Patrick Henry Winston & Berthold Klaus Paul Horn (BPB)

Additional Reading:

Programming Languages: Principles and Paradigms, M. Gabbrielli, S. Martini, Springer, ISBN: 9781848829138

CS-201: Digital Image Processing

Syllabus: **[Total Lectures: 48]**

UNIT 1. Introduction [3]

- What is Digital Image Processing?
- The origins of Digital Image Processing
- Examples of Fields that use Digital Image Processing
  - Gamma-Ray Imaging
  - X-Ray Imaging
  - Imaging in the Ultraviolet Band
  - Imaging in the Visible and Infrared Bands
  - Imaging in the Microwave Band
  - Imaging in the Radio Band
- Fundamental steps in Digital Image Processing
- Components of an Image Processing System

UNIT 2. Digital Image Fundamentals [6]

- Elements of Visual Perception
- Light and the Electromagnetic Spectrum
- Image sensing and Acquisition
- Image Sampling and Quantization
- Some Basic Relationships between Pixels
- An Introduction to the Mathematical Tools Used in Digital Image Processing
  - Array versus Matrix Operations
  - Linear versus Nonlinear Operations
  - Arithmetic Operations
  - Set and Logical Operations

UNIT 3. Intensity Transformation and Spatial Filtering [7]

- Background
- Some Basic Intensity Transformation Functions
- Histogram Processing
  - Histogram Equalization
  - Histogram Matching (Specification)
  - Local Histogram Processing
- Fundamentals of Spatial Filtering
- Smoothing Spatial Filters
- Sharpening Spatial Filters
- Combining Spatial Enhancement Methods

UNIT 4. Filtering in the Frequency Domain [10]

- Background
- Preliminary Concepts
- Sampling and the Fourier Transform of Sampled Functions

- The Discrete Fourier Transform (DFT) of One variable
- Extension to Functions of Two Variables
- Some Properties of the 2-D Discrete Fourier Transform
- The Basics of Filtering in the Frequency Domain
- Image Smoothing Using Frequency Domain Filters
- Image Sharpening Using Frequency Domain Filters
- Selective Filtering

UNIT 5. Image Restoration and Reconstruction [6]

- A Model of the Image Degradation / Restoration Process
- Noise Models
- Restoration in the Presence of Noise Only- Spatial Filtering
- Periodic Noise Reduction by Frequency Domain Filtering
    - Bandreject Filters
    - Bandpass Filters
    - Notch Filters
- Estimating the Degradation Function
- Inverse Filtering
- Minimum Mean Square Error(Wiener) Filtering
- Geometric Mean Filter

UNIT 6. Morphological Image Processing [5]

- Preliminaries
- Erosion and Dilation
- Opening and Closing
- The Hit-or-Miss Transformation
- Some Basic Morphological Algorithms
    - Boundary Extraction
    - Hole Filling
    - Extraction of Connected Components
    - Convex Hull
    - Thinning
    - Thickening
    - Skeletons
    - Pruning
    - Morphological Reconstruction

UNIT 7. Image Segmentation [7]

- Fundamentals
- Point, Line, and Edge Detection
    - Background
    - Detection of Isolated Points
    - Line Detection
    - Edge Models
    - Basic Edge Detection
    - Edge Linking and Boundary Detection
- Thresholding

        - Foundation
        - Basic Global Thresholding
        - Optimum Global Thresholding Using Otsu's Method
        - Using Image Smoothing to Improve Global Thresholding
        - Using Edges to Improve Global Thresholding
    - Region-Based Segmentation

UNIT 8. Representation and Description [4]

- Representation
    - Boundary (Border) Following
    - Chain Codes
    - Polygonal Approximations Using Minimum-Perimeter Polygons
    - Other Polygonal Approximation Approaches
    - Signatures
    - Boundary Segments
    - Skeletons
- Boundary Descriptors
    - Some Simple Descriptors
    - Shape Numbers
    - Fourier Descriptors
- Regional Descriptors
    - Some Simple Descriptors
    - Topological Descriptors
    - Texture

Text Book:

1. Gonzalez, R. C. and Woods, R. E. [2002/2008], Digital Image Processing, 3rd ed., Prentice Hall

Reference Books:

1. Sonka, M., Hlavac, V., Boyle, R. [1999]. Image Processing, Analysis and Machine Vision (2nd edition), PWS Publishing, or (3rd edition) Thompson Engineering, 2007

2. Gonzalez, R. C., Woods, R. E., and Eddins, S. L. [2009]. Digital Image Processing Using MATLAB, 2nd ed., Gatesmark Publishing, Knoxville, TN.

3. Anil K. Jain [2001], Fundamentals of digital image processing (2nd Edition), Prentice-Hall, NJ

4. Willian K. Pratt [2001], Digital Image Processing (3rd Edition), , John Wiley & Sons, NY

5. Burger, Willhelm and Burge, Mark J. [2008]. Digital Image Processing: An Algorithmic Introduction Using Java, Springer

6. Digital Image Analysis (With CD-ROM), Kropatsch, Springer, ISBN 978038795066

7. Digital Image Processing, 6e (With CD), Jähne, Springer, ISBN:978-3-540-24035-8 2