JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech (CSE) – III-II Sem

L	T	P	C
3	0	0	3

(19A05603a) COMPILER DESIGN
(Professional Elective-II)

COURSE OBJECTIVES:

The Course is designed to:

- Understand the System Programming concepts viz. assemblers, loaders, linkers and editors
- Introduce the basic principles of the compiler construction
- Explain the Concept of Context Free Grammars, Parsing and various Parsing Techniques.
- Explore the process of intermediate code generation.
- Illustrate the process of Code Generation and various Code optimization techniques.

Unit-I:

Introduction to Systems Software: Basic Assembler functions, Machine Dependant Assembler features, Machine Independent Assembler features, Basic Loader functions, Machine Dependant Loader features, Machine Independent Loader features, Text Editors, Language processors, The Structure of a Compiler.

A Simple Syntax-Directed Translator: Introduction, Syntax Definition, Syntax-Directed Translation, Parsing, A Translator for Simple Expressions, Lexical Analysis, Symbol Tables, Intermediate Code Generation.

Learning Outcomes:

- Recognize the importance of Systems software (L1)
- Identify the phases of a Compiler (L3)
- Outline the syntax rules (L2)

Unit-II:

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, The Lexical-Analyzer Generator Lex, Finite Automata, From Regular Expressions to Automata, Design of a Lexical-Analyzer Generator, Optimization of DFA-Based Pattern Matchers.

Learning Outcomes

- Identify the tokens in a program. (L3)
- Explain the process of lexical analysis (L2)

Unit – III:

Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar, Writing a Grammar, Bottom-Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers, Using Ambiguous Grammars, Parser Generators.

Learning Outcomes

- Examine the syntax of program constructs (L4)
- Evaluate the correctness of a program (L5)

Unit – IV:

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation orders for SDD's, Application of SDT, SDT schemes, Implementing L-attribute SDD's.

Intermediate Code Generation: Variants of Syntax Trees, Three address code, Translation of Expressions, Control Flow

Learning Outcomes

- Explain the process of syntax directed translation (L1)
- Develop intermediate code (L6)

Unit-V:

Code Generation: Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code, A Simple Code Generator, Peephole Optimization, Register Allocation and Assignment, Instruction Selection by Tree Rewriting, Optimal Code Generation for Expression, Dynamic Programming Code-Generation, The Principal Sources of Optimizations.

Learning Outcomes

- Generate code (L6)
- Create optimized code (L6)

Course Outcomes:

Students will be able to:

- Differentiate the various phases of a compiler (L4).
- Identify the tokens and verify the code (L4)
- Design code generator (L6)
- Apply code optimization techniques (L3)
- Design a compiler for a small programming language (L6)

Text Books :

1. Leland L. Beck, “System Software – An Introduction to Systems Programming”,3rd Edition, Pearson Education Asia, 2008.
2. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, “Compilers Principles, Techniques and Tools”, 2nd Edition, Pearson.

Reference Books

1. Yunlin Su, Song Y. Yan, “Principles of Compilers”, Springer, 2012.
2. Andrew W. Appel, “Modern Compiler Implementation in JAVA”, 2nd edition, Cambridge University Press, 2004.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech (CSE) – III-II Sem

L	T	P	C
3	0	0	3

(19A05603b) INTRODUCTION TO MACHINE LEARNING

Professional Elective-II

(Common to CSE & IT)

Course Objectives:

This course is designed to:

- Understand the basic theory underlying machine learning
- Formulate machine learning problems corresponding to different applications.
- Illustrate a range of machine learning algorithms along with their strengths and weaknesses
- Apply machine learning algorithms to solve problems of moderate complexity.
- Understand how Machine Learning imbibes the philosophy of Human learning.

UNIT I

Introduction: Learning Problems – Perspectives and Issues – Concept Learning – Version Spaces and Candidate Eliminations – Inductive bias – Decision Tree learning – Representation – Algorithm – Heuristic Space Search.

Learning Outcomes:

At the end of the unit, students will be able to:

- Explore how to build computer programs that improve their performance at some task through experience. (L6).
- Interpret Decision tree learning as practical methods for inductive inference. (L2)

UNIT II

NEURAL NETWORKS AND GENETIC ALGORITHMS: Neural Network Representation – Problems – Perceptrons – Multilayer Networks and Back Propagation Algorithms – Advanced Topics – Genetic Algorithms – Hypothesis Space Search – Genetic Programming – Models of Evolution and Learning.

Learning Outcomes:

At the end of the unit, students will be able to:

- Appraise artificial neural networks as one of the most effective learning methods currently known to interpret complex real-world sensor data,. (L5).
- Illustrates the use of the genetic algorithm approach, and examine the nature of its hypothesis space search.(L2)

UNIT III

BAYESIAN AND COMPUTATIONAL LEARNING: Bayes Theorem – Concept Learning – Maximum Likelihood – Minimum Description Length Principle – Bayes Optimal Classifier – Gibbs Algorithm – Naïve Bayes Classifier – Bayesian Belief Network – EM Algorithm – Probability Learning – Sample Complexity – Finite and Infinite Hypothesis Spaces – Mistake Bound Model.

Learning Outcomes:
At the end of the unit, students will be able to:

- Illustrate the principles of Probability for classification as an important area of Machine Learning Algorithms. (L2)
- Analyze sample complexity and computational complexity for several learning Problems (L4)

UNIT IV

INSTANCE BASED LEARNING: K- Nearest Neighbor Learning – Locally weighted Regression – Radial Bases Functions – Case Based Learning.

Learning Outcomes:
At the end of the unit, students will be able to:

- Infer that the Instance based algorithms can be used to overcome memory complexity and overfitting problems. (L2).

UNIT V

ADVANCED LEARNING : Learning Sets of Rules – Sequential Covering Algorithm – Learning Rule Set – First Order Rules – Sets of First Order Rules – Induction on Inverted Deduction – Inverting Resolution – Analytical Learning – Perfect Domain Theories – Explanation Base Learning – FOCL Algorithm – Reinforcement Learning – Task – Q-Learning – Temporal Difference Learning

Learning Outcomes:
At the end of the unit, students will be able to:

- Infer that the combined methods outperform both purely inductive and purely analytical learning methods. (L2)
- Recognize the importance of Reinforcement Learning in the industry.

Course Outcomes:

Upon completion of the course, the students should be able to:

- Identify machine learning techniques suitable for a givenproblem. (L3)

- Solve the real world problems using various machine learningtechniques. (L6)
- Apply Dimensionality reductiontechniques for data preprocessing. (L3)
- Explain what is learning and why it is essential in the design of intelligent machines. (L2)
- Implement Advanced learning models for language, vision, speech, decision making etc. (L1)

Text Books:

1) T.M. Mitchell, "Machine Learning", McGraw-Hill,1997.

Reference Books:

1) Ethern Alpaydin, "Introduction to Machine Learning", MIT Press,2004.
2) Stephen Marsland, "Machine Learning -An Algorithmic Perspective", Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series,2014.
3) Andreas C. Müller and Sarah Guido "Introduction to Machine Learning with Python:A Guide for Data Scientists",Oreilly.

e-Resources:

1) Andrew Ng, "Machine Learning Yearning"https://www.deeplearning.ai/machine-learning- yearning/
2) Shai Shalev-Shwartz , Shai Ben-David, "Understanding Machine Learning: From Theory to Algorithms" , Cambridge University Press https://www.cse.huji.ac.il/~shais/UnderstandingMachineLearning/index.html

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech (CSE) – III-II Sem **L T P C**

3 0 0 3

(19A05603c) REAL TIME SYSTEMS

Professional Elective-II

(Common to CSE & IT)

Course Objectives:

The Course is designed to:

- Understand the requirements of Real Time Operating Systems.
- Illustrate Real Time features using case studies.
- Describe how a real-time operating system kernel is implemented.

UNIT – I

Introduction: Introduction to UNIX/LINUX, Overview of Commands, File I/O,(open, create, close, lseek, read, write), Process Control (fork, vfork, exit, wait, waitpid, exec).

UNIT - II

Real Time Operating Systems: Brief History of OS, Defining RTOS, The Scheduler, Objects, Services, Characteristics of RTOS, Defining a Task, asks States and Scheduling, Task Operations, Structure, Synchronization, Communication and Concurrency. Defining Semaphores, Operations and Use, Defining Message Queue, States, Content, Storage, Operations and Use

UNIT - III

Objects, Services and I/O: Pipes, Event Registers, Signals, Other Building Blocks, Component

Configuration, Basic I/O Concepts, I/O Subsystem

UNIT - IV

Exceptions, Interrupts and Timers: Exceptions, Interrupts, Applications, Processing of Exceptions and Spurious Interrupts, Real Time Clocks, Programmable Timers, Timer Interrupt Service Routines (ISR), Soft Timers, Operations.

UNIT - V

Case Studies of RTOS: RT Linux, MicroC/OS-II, Vx Works, Embedded Linux, and Tiny OS.

Course Outcomes:

Students will be able to:

1. Explain real-time concepts such as preemptive multitasking, task priorities, priority inversions, mutual exclusion, context switching, and synchronization, interrupt latency and response time, and semaphores. (L2)
2. Describe how tasks are managed. (L1)
3. Discuss how tasks can communicate using semaphores, mailboxes, and queues. (L6)
4. Build a real-time system on an embedded processor.(L6)
5. Examine the real time operating systems like RT Linux, Vx Works, MicroC /OSII, Tiny OS (L4)

TEXT BOOK:

1. Real Time Concepts for Embedded Systems – Qing Li, Elsevier, 2011

REFERENCE BOOKS:

1. Embedded Systems- Architecture, Programming and Design by Rajkamal, 2007, TMH.
2. Advanced UNIX Programming, Richard Stevens
3. Embedded Linux: Hardware, Software and Interfacing – Dr. Craig Hollabaugh

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech (CSE) – III-II Sem **L T P C**

3 0 0 3

(19A05603d) ADVANCED COMPUTER ARCHITECTURE

Professional Elective-II

(Common to CSE & IT)

Prerequisites: Computer Organization

Course Objectives

The Course is designed to:

- Impart the concepts and principles of parallel and advanced computer architectures.
- Develop the design techniques of Scalable and multithreaded Architectures.
- Apply the concepts and techniques of parallel and advanced computer architectures to design modern computer systems

UNIT - I

Theory of Parallelism, Parallel computer models, The State of Computing, Multiprocessors and
Multicomputers, Multivector and SIMD Computers, PRAM and VLSI models, Architectural development tracks, Program and network properties, Conditions of parallelism, Program partitioning and Scheduling, Program flow Mechanisms, System interconnect Architectures.

UNIT - II

Principals of Scalable performance, Performance metrics and measures, Parallel Processing applications, Speed up performance laws, Scalability Analysis and Approaches, Hardware Technologies, Processes and Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors, Memory Hierarchy Technology, Virtual Memory Technology.

UNIT - III

Bus Cache and Shared memory, Backplane bus systems, Cache Memory organizations, Shared-
Memory Organizations, Sequential and weak consistency models, Pipelining and superscalar techniques, Linear Pipeline Processors, Non-Linear Pipeline Processors, Instruction Pipeline design, Arithmetic pipeline design, superscalar pipeline design.

UNIT - IV

Parallel and Scalable Architectures, Multiprocessors and Multicomputers, Multiprocessor system
interconnects, cache coherence and synchronization mechanism, Three Generations of

Multicomputers, Message-passing Mechanisms, Multivetor and SIMD computers, Vector Processing Principals, Multivector Multiprocessors, Compound Vector processing, SIMD computer Organizations, The connection machine CM-5

UNIT - V

Scalable, Multithreaded and Dataflow Architectures, Latency-hiding techniques, Principals of Multithreading, Fine-Grain Multicomputers, Scalable and multithreaded Architectures, Dataflow and hybrid Architectures.

Course Outcomes:

Students will be able to:

- Explain Computational models and Computer Architectures.(L2)
- Elaborate the Concepts of parallel computer models.(L6)
- Define Scalable Architectures, Pipelining, Superscalar processors, multiprocessors (L1)

TEXT BOOK:

1. Advanced Computer Architecture Second Edition, Kai Hwang, Tata McGraw Hill Publishers.

REFERENCE BOOKS:

1. Computer Architecture, Fourth edition, J. L. Hennessy and D.A. Patterson. ELSEVIER.

R18 B.Tech. CSE Syllabus JNTU HYDERABAD

2. Advanced Computer Architectures, S.G. Shiva, Special Indian edition, CRC, Taylor &Francis.
3. Introduction to High Performance Computing for Scientists and Engineers, G. Hager and G. Wellein, CRC Press, Taylor & Francis Group.
4. Advanced Computer Architecture, D. Sima, T. Fountain, P. Kacsuk, Pearson education.
5. Computer Architecture, B. Parhami, Oxford Univ. Press.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech (CSE) – III-II Sem

L	T	P	C
3	0	0	3

(19A05603e) COMPUTER VISION

Professional Elective-II

(Common to CSE & IT)

COURSE OBJECTIVES:

The Course is designed to:

- Understand shape and region analysis
- Illustrate Hough Transform and its applications to detect lines, circles, ellipses
- Explain three-dimensional image analysis techniques
- Describe motion analysis
- Study some applications of computer vision algorithms

UNIT I IMAGE PROCESSING FOUNDATIONS:

Review of image processing techniques – classical filtering operations – thresholding techniques – edge detection techniques – corner and interest point detection – mathematical morphology – texture

UNIT II SHAPES AND REGIONS:

Binary shape analysis – connectedness – object labeling and counting – size filtering – distance functions – skeletons and thinning – deformable shape analysis – boundary tracking procedures – active contours – shape models and shape recognition – centroidal profiles – handling occlusion – boundary length measures – boundary descriptors – chain codes – Fourier descriptors – region descriptors – moments

UNIT III HOUGH TRANSFORM:

Line detection – Hough Transform (HT) for line detection – foot-of-normal method – line localization – line fitting – RANSAC for straight line detection – HT based circular object detection – accurate center location – speed problem – ellipse detection – Case study: Human Iris location – hole detection – generalized Hough Transform (GHT) – spatial matched filtering – GHT for ellipse detection – object location – GHT for feature collation

UNIT IV 3D VISION AND MOTION:

Methods for 3D vision – projection schemes – shape from shading – photometric stereo – shape from texture – shape from focus – active range finding – surface representations – point-based representation – volumetric representations – 3D object recognition – 3D reconstruction – introduction to motion – triangulation – bundle adjustment – translational alignment – parametric motion – splinebased motion – optical flow – layered motion

UNIT V APPLICATIONS:

Application: Photo album – Face detection – Face recognition – Eigen faces – Active appearance and 3D shape models of faces Application: Surveillance – foreground-background separation – particle filters – Chamfer matching, tracking, and occlusion – combining views from multiple cameras – human gait analysis Application: In-vehicle vision system: locating roadway – road markings – identifying road signs – locating pedestrians

Course Outcomes:

Students will be able to:

- Apply fundamental image processing techniques required for computer vision (L3)
- Illustrate shape analysis (L2)
- Evaluate boundary tracking techniques (L5)
- Apply chain codes and other region descriptors (L3)
- Apply 3D vision techniques (L3)
- Develop applications using computer vision techniques (L6)

REFERENCES:

1. D. L. Baggio et al., “Mastering OpenCV with Practical Computer Vision Projects”, Packt Publishing, 2012.

2. E. R. Davies, “Computer & Machine Vision”, Fourth Edition, Academic Press, 2012.

3. Jan Erik Solem, “Programming Computer Vision with Python: Tools and algorithms for analyzing images”, O'Reilly Media, 2012.

4. Mark Nixon and Alberto S. Aquado, “Feature Extraction & Image Processing for Computer Vision”, Third Edition, Academic Press, 2012.

5. R. Szeliski, “Computer Vision: Algorithms and Applications”, Springer 2011. 6. Simon J. D. Prince, “Computer Vision: Models, Learning, and Inference”, Cambridge University Press, 2012.