UNIVERSITY OF KERALA

B. TECH. DEGREE COURSE

(2013 SCHEME)

SYLLABUS FOR

VI SEMESTER

COMPUTER SCIENCE & ENGINEERING

SCHEME -2013

VI SEMESTER

COMPUTER SCIENCE & ENGINEERING (R)

Course No	Name of subject	Credits	Weekly load, hours			C A Marks	Exam Duration Hrs	U E Max Marks	Total Marks
			L	T	D/ P				
13.601	Compiler Design (FR)	4	3	1	-	50	3	100	150
13.602	Principles of Programming Languages (R)	3	2	1	-	50	3	100	150
13.603	Design and Analysis of Algorithms (FR)	4	3	1	-	50	3	100	150
13.604	Computer Networks (FR)	3	2	1	-	50	3	100	150
13.605	PC Hardware and Interfacing (R)	4	3	1	-	50	3	100	150
13.606	Signals and Systems (R)	3	2	1	-	50	3	100	150
13.607	Microprocessor Lab (R)	4	-	-	4	50	3	100	150
13.608	System Software Lab (R)	4	-	-	4	50	3	100	150
	Total	**29**	**15**	**6**	**8**	**400**		**800**	**1200**

13.601 COMPILER DESIGN (FR)

Teaching Scheme: 3(L) - 1(T) - 0(P) **Credits**: 4

Course Objective:

- *To introduce the major concept areas of language translation and compiler design*
- *To develop an awareness of the function and complexity of modern compilers.*
- *To provide practical, hands on experience in compiler design.*

Pre-requisites: ***13.306*** - *Data Structures and Algorithms*,

13.504 - *System Programming*

Module – I

Introduction to compilers and interpreters – Overview of compilation, Issues in compilation – structure of a compiler – compiler writing tools – bootstrapping – notations and concepts for languages and grammars – regular expressions – context free grammar, derivations and parse trees, BNF notations.

Module – II

Context of a lexical analyzer – construction of lexical analyzer, deterministic and non deterministic finite automata. Compile time error handling, error detection, reporting, recovery and repair.

Module – III

Basic parsing techniques – Top down parsing – recursive descent parser, predictive parser simple LL(1) grammar. Bottom up parsers, operator precedence parser, LR grammar, LR(0), SLR(1), LALR(1) parsers.

Module – IV

Syntax directed translation schemes, intermediate codes, translation of assignments, translation of array reference, Boolean expressions, case statements, back patching. Code optimization, loop optimization and global optimization, sources of sample code generation.

References:

1. Aho A. V., M. S. Lam, R. Sethi and J. D. Ullman, *Compilers: Principles, Techniques and Tools*, 2nd Edn., Pearson Education.
2. Keith D Cooper and Linda Torczon, *Engineering a Compiler*, 2nd Edn, Elsevier.
3. Andrew W. Appel, *Modern Compiler Implementation in C,* Cambridge University Press.

4. Kenneth C. Louden, *Compiler Construction: Principles and Practice,* Cengage Learning.
5. Kakde O. G., *Algorithms for Compiler Design*, Cengage Charles River Media.
6. Raghavan V., *Principles of Compiler Design*, TMH.

Internal Continuous Assessment *(Maximum Marks-50)*

50% - Tests (minimum 2)
30% - Assignments (minimum 2) such as home work, problem solving, quiz, literature survey, seminar, term-project, software exercises, etc.
20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours Maximum Total Marks: 100

The question paper shall consist of 2 parts.

Part A (20 marks) - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

Part B (80 Marks) - Candidates have to answer one full question out of the two from each module. Each question carries 20 marks.

Note: *The question paper shall contain at least 30% analytical/problem solving questions.*

Course Outcome:

After successful completion of this course, the students will be able to:

- *Identify different language translators and explain the concepts and different phases of compilation with compile time error handling.*
- *Represent language tokens using regular expressions, context free grammar and finite automata and design lexical analyzer for a language.*
- *Compare top down with bottom up parsers, and develop appropriate parser to produce parse tree representation of the input.*
- *Explain syntax directed translation schemes for a given context free grammar and generate intermediate code.*
- *Apply optimization techniques to intermediate code and generate machine code for high level language program.*

13.602: PRINCIPLES OF PROGRAMMING LANGUAGES (R)

Teaching Scheme: 2(L) - 1(T) - 0(P) **Credits**: 3

Course Objectives:

- *To improve the background for choosing appropriate programming languages for certain classes of programming problems*
- *To improve Increase the ability to learn new programming languages*
- *Increase the capacity to express programming concepts and choose among alternative ways to express things.*

Pre-requisites: ***13.109*** *Foundations of Computing and Programming in C*
13.403 *Object Oriented Techniques*
13.506 *Object oriented Design & JAVA Programming*

Module – I

Names, Scopes, and Bindings:-Names and Scopes, Binding Time, Scope Rules, Storage Management, Aliases, Overloading, Polymorphism, Binding of Referencing Environments.

Control Flow: - Expression Evaluation, Structured and Unstructured Flow, Sequencing, Selection, Iteration, Recursion, Nondeterminacy.

Data Types:-Type Systems, Type Checking, Records and Variants, Arrays, Strings, Sets, Pointers and Recursive Types, Lists, Files and Input/Output, Equality Testing and Assignment.

Module – II

Subroutines and Control Abstraction: - Static and Dynamic Links, Calling Sequences, Parameter Passing, Generic Subroutines and Modules, Exception Handling, Coroutines.

Functional and Logic Languages:-Lambda Calculus, Overview of Scheme, Strictness and Lazy Evaluation, Streams and Monads, Higher-Order Functions, Logic Programming in Prolog, Limitations of Logic Programming.

Module – III

Data Abstraction and Object Orientation:-Encapsulation, Inheritance, Constructors and Destructors, Dynamic Method Binding, Multiple Inheritance.

Innovative features of Scripting Languages:-Scoping rules, String and Pattern Manipulation, Data Types, Object Orientation.

Module – IV

Concurrency:- Threads, Synchronization, Language-Level Mechanisms.

Run-time program Management:- Virtual Machines, Late Binding of Machine Code, Reflection, Symbolic Debugging, Performance Analysis.

References

1. Scott M. L., *Programming Language Pragmatics*, 3rd Edn., Morgan Kaufmann Publishers.
2. Kenneth C. Louden, *Programming Languages: Principles and Practice*, 2nd Edn., Cengage Learning.
3. Tucker A. B. and R. E. Noonan, *Programming Languages: Principles and Paradigms*, 2nd Edn. –TMH.
4. R.W. Sebesta, *Concepts of Programming Languages*, 8th Edn., Pearson Education.
5. Ravi Sethi, *Programming Languages: Concepts & Constructs*, 2nd Edn., Pearson Education
6. David A. Watt, *Programming Language Design Concepts,* Wiley Dreamtech.
7. Pratt T. W., M. V. Zelkowitz, and T. V. Gopal, *Programming Languages: Design and Implementation*, 4th Edn., Pearson Education.
8. Ghezzi C. and M. Jazayeri, *Programming Language Concepts*, 3rd Edn, Wiley.

Internal Continuous Assessment *(Maximum Marks-50)*

50% - Tests (minimum 2)

30% - Assignments (minimum 2) such as home work, problem solving, literature survey, seminar, term-project, software exercises, etc.

20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours *Maximum Total Marks: 100*

The question paper shall consist of 2 parts.

Part A (20 marks) - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

Part B (80 Marks) - Candidates have to answer one full question (question may contain sub-divisions), out of the two from each module. Each question carries 20 marks.

Course outcome:

At the end of the course, the student will be able to

- *Use language concepts related to data, operations, sequence control, data control and storage management to select a suitable programming language for an application.*

- *Use the knowledge of concepts of syntax and semantics of language features along with their internal implementation details to design a new programming language suitable for a specific application domain.*
- *Explain advanced language constructs used in functional, object oriented and logic programming languages.*
- *Describe run time program management and innovative features of scripting language.*

13.603 DESIGN AND ANALYSIS OF ALGORITHMS (FR)

Teaching Scheme: 3(L) - 1(T) - 0(P) **Credits**: 4

Course Objectives:

- *Analyze the asymptotic performance of algorithms.*
- *Write rigorous correctness proofs for algorithms.*
- *Demonstrate a familiarity with major algorithms and data structures.*
- *Apply important algorithmic design paradigms and methods of analysis.*
- *Synthesize efficient algorithms in common engineering design situations.*

Pre-requisites: ***13.306 -*** *Data Structures and Algorithms.*

Module – I

Concepts in algorithm analysis – the efficiency of algorithms, average and worst – case analysis, Asymptotic notation, time and space complexity, Recurrences – substitution method, iteration method and master method, Analysis of sorting algorithms – insertion sorting, heaps, maintaining the heap property, building heap, heap sort algorithm, priority queues. Description of quick sort, randomised version of quick sort.

Module – II

Height balanced trees – AVL TREES – Red-Black trees – Steps involved in insertion and deletion – rotations, Definition of B-trees – basic operations on B-trees, Algorithm for sets – Union and Find operations on disjoint sets.

Module – III

Graphs – DFS and BFS traversals, Spanning trees – Minimum Cost Spanning Trees, Kruskal's and Prim's algorithms, Shortest paths – single source shortest path algorithms, Topological sorting, strongly connected components. Algorithm Design and analysis Techniques – Divide and Conquer techniques – Merge Sort, Integer multiplication problem, Strassen's algorithm.

Module – IV

Dynamic programming – Matrix multiplication problem, Greedy algorithms – Knapsack problem, Back tracking – 8 Queens problem, Branch and Bound – Travelling Salesman problem. Definitions and Basic concepts of NP-completeness and NP-Hardness. Study of NP Complete problems.

References:

1. Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest, *Introduction to Algorithms,* PHI.

2. Horowitz and Sahni, *Fundamentals of Computer Algorithms*, Galgotia Publication.
3. Kenneth A. Merman and Jerome L. Paul, *Fundamentals of Sequential and Parallel Algorithms,* Vikas Publishing Company.

Internal Continuous Assessment *(Maximum Marks-50)*

50% - Tests (minimum 2)

30% - Assignments (minimum 2) such as home work, problem solving, quiz, literature survey, seminar, term-project, software exercises, etc.

20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours *Maximum Total Marks: 100*

The question paper shall consist of 2 parts.

Part A (20 marks) - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

Part B (80 Marks) - Candidates have to answer one full question (question may contain sub-divisions), out of the two from each module. Each question carries 20 marks.

Note: *The question paper shall contain at least 30% analytical/problem solving questions.*

Course Outcome:

After successful completion of this course, the student will be able to:

- *Define asymptotic notations to analyze the performance of algorithms. Apply substitution method, iteration method and master method to analyze recursive algorithms.*
- *Analyze and compare performance of sorting algorithms in terms of time and space complexities.*
- *Discuss various operations of Height-balanced trees and analyze performance of the operations.*
- *Illustrate various applications of graphs such as minimum cost spanning tree, shortest path, topological sorting and strongly connected components, and determine their time and space complexities.*
- *Apply different algorithm design paradigms such as divide-and conquer, dynamic programming and the greedy methods to design efficient algorithms for real world problems.*
- *Use the concepts of NP-Completeness and NP-Hardness to identify whether a given problem is tractable or not.*

13.604 COMPUTER NETWORKS (FR)

Teaching Scheme: 2(L) - 1(T) - 0(P) **Credits**: 3

Course Objective:

- *Build an understanding of the fundamental concepts of computer networking.*
- *Familiarize the student with the basic taxonomy and terminology of the computer networking area.*
- *Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.*
- *Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.*

Pre-requisites: ***13.404*** - *Data Communication*

Module – I

Introduction – Uses – Network Hardware – LAN –MAN – WAN, Internetworks – Network Software – Protocol hierarchies – Design issues for the layers – Interface & Service – Service Primitives. Reference models – OSI – TCP/IP. Data Link layer Design Issues – Flow Control and ARQ techniques. Data link Protocols – HDLC. DLL in Internet.

Module – II

MAC Sub layer – IEEE 802 FOR LANs & MANs, IEEE 802.3, 802.4, 802.5. Bridges - Switches – High Speed LANs - Gigabit Ethernet. Wireless LANs - 802.11 a/b/g/n, 802.15. Network layer – Routing – Shortest path routing, Flooding, Distance Vector Routing, Link State Routing, RIP, OSPF, Routing for mobile hosts.

Module – III

Congestion control algorithms – QoS. Internetworking – Network layer in internet. IP Addressing – Classless and Classful Addressing. Subnetting, Internet Control Protocols – ICMP, ARP, RARP, BOOTP. Internet Multicasting – IGMP, Exterior Routing Protocols – BGP . IPv6 – Addressing – Issues.

Module – IV

Transport Layer – TCP & UDP. Application layer –DNS, Electronic mail, MIME, SNMP. Introduction to World Wide Web. VoIP - H.323, SIP standards, Gatekeeper.

References:

1. Andrew S. Tanenbaum, *Computer Networks*, 4/e, PHI.
2. Behrouz A. Forouzan, *Data Communications and Networking*, 4/e, Tata McGraw Hill.

Internal Continuous Assessment *(Maximum Marks-50)*

50% - Tests (minimum 2)

30% - Assignments (minimum 2) such as home work, problem solving, quiz, literature survey, seminar, term-project etc.

20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours *Maximum Total Marks: 100*

The question paper shall consist of 2 parts.

Part A (20 marks) - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

Part B (80 Marks) - Candidates have to answer one full question (question may contain sub-divisions), out of the two from each module. Each question carries 20 marks.

Note: *The question paper shall contain at least 30% analytical/problem solving questions.*

Course Outcome:

After the successful completion of the course students will be able to:

- *Describe the different aspects of networks, protocols and network design models.*
- *Explain the various Data Link layer design issues and Data Link protocols*
- *Analyze and compare different LAN protocols*
- *Compare and select appropriate routing algorithms for a network.*
- *Describe the important aspects and functions of network layer, transport layer and application layer in internetworking.*

13.605 PC HARDWARE AND INTERFACING (R)

Teaching Scheme: 3(L) - 1(T) - 0(P) **Credits:** 4

Course Objectives:

- *To impart knowledge on the architecture and memory management of advanced Intel Microprocessors*
- *To get a detailed idea about the hardware components of a PC*
- *To impart knowledge on the PC hardware standards and technologies.*
- *To acquire knowledge on the interfacing concepts and the data acquisition through I/O ports of a PC.*

Pre-requisites: ***13.402*** *- Computer Organization and Design*
13.505 *- Microprocessors and Interfacing*

Module – I

Advanced Microprocessors- Intel 80286- Internal Architecture, modes of operation, 80386 – Internal Architecture, modes of operation, memory management, segmentation and paging, comparison with 80486, Intel Pentium- Internal Architecture, Branch Prediction, memory management, Intel Pentium4- Net Burst micro architecture.

Module – II

Hardware organization of PC, Mother board- mother board controllers and system resources-memory address, I/O port, IRQ, Chipsets- North bridge, South Bridge, ROM BIOS, ROM POST, Physical Form Factors-AT and ATX.

Memory- Memory organization- conventional, extended and expanded memory, Static and Dynamic RAM, Memory chips and modules, Advanced Memory Technologies- RDRAM, DDRAM, PPRAM.

Module – III

Power Supply- Power supply control signals, Ventilation and Cooling Protection- Power Supply Fan, Processor Cooling.

Mass Storage Interface- IDE Interface- ATA standards, Data Transfer Modes, SCSI Interface- SCSI standards, SCSI Hardware.

Magnetic Storage- Reading and Writing data, Hard Disk drives-components.

Optical Storage- Optical Storage Media, CDROM drives, Recordable Drives. DVD ROM drives – DVD drive and decoder.

Module – IV

I/O ports and Devices- Standard Parallel Port- Registers, Interface to SPP, Data Acquisition through Parallel Port, Bidirectional operation, Simple programming examples.

Serial Port- pins and signals, UART, Registers, Data Acquisition through Serial Port.

I/O Buses- ISA, MCA, EISA, PCI, AGP.

References:

1. Douglas V. Hall, *Microprocessors and Interfacing-Programming and Hardware,* McGraw Hill. (Module I).
2. Ray A K, K M Bhurchandi, *Advanced Microprocessors and Peripherals*, McGraw Hill, 2nd Edition (Module I).
3. Craig Zacker, John Rourke, *The Complete Reference –PC Hardware,* Tata McGraw Hill (Modules I, II, III).
4. Mathvanan N., *Microprocessors, PC Hardware and Interfacing*, PHI Learning Pvt. Ltd. (Module IV).
5. Barry B Brey, The Intel Microprocessors-Architecture, Programming and Interfacing, Pearson (Module I)

Internal Continuous Assessment *(Maximum Marks-50)*

50% - Tests (minimum 2)

30% - Assignments (minimum 2) such as class room/home work, problem solving, quiz, literature survey, seminar, term-project, software exercises, etc.

20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours *Maximum Total Marks: 100*

The question paper shall consist of 2 parts.

Part A (20 marks) - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

Part B (80 Marks) - Candidates have to answer one full question (question may contain sub-divisions), out of the two from each module. Each question carries 20 marks.

Note: *The question paper shall contain at least 30% analytical/problem solving questions.*

Course Outcome:

After successful completion of this course, students will be able to:

- *Identify the basic elements and functions of contemporary microprocessors*
- *Identify types and characteristics of various peripherals, including storage and I/O interfaced with advanced microprocessors*
- *Design of external hardware interfacing circuit for a PC system*
- *Gain knowledge in doing data acquisition experiments on PCs by using parallel port and serial port.*

13.606 SIGNALS AND SYSTEMS (R)

Teaching Scheme: 2(L) - 1(T) - 0(P) **Credits**: 3

Course Objective:

- *Coverage of continuous and discrete-time signals and systems, their properties and representations and methods that are necessary for the analysis of continuous and discrete-time signals and systems.*
- *Knowledge of time-domain representation and analysis concepts as they relate to difference equations, impulse response and convolution, etc.*
- *Knowledge of frequency-domain representation and analysis concepts using Fourier Analysis tools, Z-transform*
- *Concepts of the sampling process*

Module – I

Basic continuous time signals, signal sampling theorem, decimation and interpolation, aliasing elementary signals and quantization, discretization of continuous time signals, discrete time signals. Types of signals and operations on signals, Classification of systems, properties of systems, Circuits – sinusoidal and periodic signal response, frequency response, transfer function, Examples of systems for controls and communication, Differential Equations – initial conditions, complete response, transient and steady-state response, zero state and zero-input response.

Module – II

Linear Time Invariant (LTI) systems: Representation of systems using differential difference equation, Impulse, step and exponential response, system stability, examples on applications of LTI systems, convolution sum and convolution integral, impulse response of interconnected systems, auto-correlation, cross correlation, properties of correlation, analogy between correlation and convolution, total response of a system

Laplace Transform: Overview of Laplace Transform: Laplace Transform and properties, relation between continuous time Fourier Transform and Laplace Transform, unilateral Laplace Transform. Analysis of continuous time LTI systems using Laplace Transform, Transfer Function, causality and stability of systems, solution of differential equation using Laplace Transform.

Module – III

Z-Transform, region of convergence, properties of Z-transform, inverse Z transform, unilateral transform, Analysis of discrete time LTI systems using z-Transform, Transfer Function, causality and stability of systems, frequency response, relation between Laplace

Transform and z–Transform. Fourier Series, properties, trigonometric and exponential Fourier series representation of signals, magnitude and phase spectra, power spectral density and bandwidth, Continuous Time Fourier Transform (CTFT) and Discrete Time Fourier Transform (DTFT), Fourier Transform and Inverse Fourier Transform on periodic and non-periodic signals, relation between discrete time Fourier Transform and z-Transform, Properties, limitations of Fourier Transform and need for Laplace and z-transform.

Module – IV

Introduction to DFT – Properties of DFT – Filtering methods based on DFT – FFT Algorithms Decimation – in – time Algorithms, Decimation – in – frequency Algorithms – Use of FFT in Linear Filtering – DCT.

Digital filter structures – block diagram and signal flow graph representation – structures for IIR – direct form structure – Cascade form structure – parallel form structure – lattice structure. Structures for FIR – direct form structures – direct form structure of linear phase system – cascade form structure – frequency sampling structure – lattice structure.

References:-

1. Bandyopadhyaya M. N., *Introduction to Signals and Systems and Digital Signal Processing,* PHI.
2. Li Tan, *Digital Signal Processing, Fundamentals and Applications,* Elsevier.
3. Oppenheim A. V. and R. W. Schafer, *Digital Signal Processing,* Prentice-Hall Inc.
4. Proakis J.K. and D. G. Manolakis, *Introduction to Digital Signal Processing,* MacMillan.
5. Hayes M. H., *Digital Signal Processing*, Tata McGraw Hill (SCHAUM's Outlines).
6. Apte S. D., *Digital Signal Processing,* Wiley India.
7. Ambardar A., *Digital Signal Processing*: A Modern Introduction, Thomson India Edition.
8. Mitra S. K., *Digital Signal Processing*, Wiley.
9. Smith S. W., *Digital Signal Processing : A Practical Guide for Engineers and Scientists*, Elsevier India
10. Ramesh Babu P., *Digital Signal Processing*, Scitech Publications.
11. Ganesh Rao D. and V. P. Gejji, *Digital Signal Processing: Theory and Lab Practice*, Sanguine Publishers.

Internal Continuous Assessment *(Maximum Marks-50)*

50% - Tests (minimum 2)

30% - Assignments (minimum 2) such as class room/home work, problem solving, quiz, literature survey, seminar, term-project, software exercises, etc.

20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours *Maximum Total Marks: 100*

The question paper shall consist of 2 parts.

Part A (20 marks) - Five Short answer questions of 4 marks each. All questions are compulsory. There should be at least one question from each module and not more than two questions from any module.

Part B (80 Marks) - Candidates have to answer one full question (question may contain sub-divisions), out of the two from each module. Each question carries 20 marks.

Note: *The question paper shall contain at least 30% analytical/problem solving questions.*

Course Outcome:

After successful completion of this course, students will be able to

- *Apply time and frequency domain analysis techniques to different types of signals and systems*
- *Classify Signals and systems as discrete/continuous, linear/non-linear, causal/non-causal, time variant/invariant etc*
- *Understand the need to define the Laplace and Z transforms to analyze a class of systems*
- *Select and utilize appropriate Fourier transform methods for basic signal processing applications.*

13.607 MICROPROCESSOR LAB (R)

Teaching Scheme: 0(L) - 0(T) - 4(P) **Credits**: 4

Course Objective :

- *To design assembly language programs for solving problems*
- *To understand organization of interfacing devices for various peripheral devices and programming them*

Pre-requisites: ***13.505*** *Microprocessors & Interfacing*

List of Exercises:

1. Study of 8086 trainer kit by executing simple programs such as code conversion, decimal arithmetic and bit manipulation
2. Study of Assembler and Debugging commands
3. Programming with 8086 – Addition of 32 bit numbers, matrix multiplication, factorial, LCM, GCD, Fibonacci, String manipulation, search, find and replace, copy operations, sorting.(PC Required)
4. Interfacing 8086 with the following and conduct experiments:
   a) 8255, 8279, 8259, and 8253/54.
   b) Stepper Motor
   c) ADC and DAC.
5. Parallel Communication between two Microprocessor Kits using Mode 1 and Mode 2 of 8255.
6. Interfacing Microprocessor kit with PC using RS 232

Internal Continuous Assessment *(Maximum Marks-50)*

40% - Test

40% - Class work and Record (Up-to-date lab work, problem solving capability, keeping track of rough record and fair record, term project, etc.)

20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours *Maximum Total Marks: 100*

Questions based on the list of exercises prescribed.

Marks should be awarded as follows:

20% - Algorithm/Design

30% - Implementing / Conducting the work assigned

25% - Output/Results and inference

25% - Viva voce

Candidate shall submit the certified fair record for endorsement by the external examiner.

Course Outcome:

After successful completion of this course, students will be able to:

- *Develop assembly language programs for problem solving*
- *Implement assembly language program to interface various I/O devices.*

13.608 SYSTEM SOFTWARES LAB (R) (R)

Teaching Scheme: 0(L) - 0(T) - 4(P) **Credits**: 4

Course Objective :

- *To design and implement assembler for a hypothetical machine.*
- *To design Macro processor.*
- *To get an exposure to design and implement various components of system software*

Pre-requisites: ***13.504*** *Systems Programming,*
13.601 *Compiler Design*

List of Exercises:

1. Design of a single pass assembler for a hypothetical Machine
2. Design of a 2 – pass assembler for a hypothetical machine
3. Design of assembler which generates code with relocation option
4. Design of absolute loader
5. Design of relocating loader
6. Design of macro processor
7. Lexical analysis
8. Operator precedence relations
9. Recursive descent parser
10. First and follow
11. Intermediate code generation
12. Code generation

Internal Continuous Assessment *(Maximum Marks-50)*

40% - Test

40% - Class work and Record (Up-to-date lab work, problem solving capability, keeping track of rough record and fair record, term projects etc.)

20% - Regularity in the class

University Examination Pattern:

Examination duration: 3 hours *Maximum Total Marks: 100*

Marks should be awarded as follows:

20% - Algorithm/Design

30% - Implementing / Conducting the work assigned

25% - Output/Results and inference

25% - Viva voce

Candidate shall submit the certified fair record for endorsement by the external examiner.

Course Outcome:

After successful completion of this course, students will be able to:

- *Understand latest features of translators.*
- *Apply the concept of finite automata to implement components of system software.*
- *Design system software using latest tools.*