

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR
(Established by Govt. of A.P., Act. No. 30 of 2008)
ANANTHAPURAMU – 515 002 (A.P) INDIA
Course Structure & Syllabus for B.Tech
R13 Regulations

COMPUTER SCIENCE & ENGINEERING

B.Tech. III - I Semester

S.No	Course code	Subject	Theory	Tu / Lab	Credits
1.	13A05501	Operating Systems	3	1 -	3
2.	13A05502	Compiler Design	3	1 -	3
3.	13A05503	Unix and Shell Programming	3	1 -	3
4.	13A05504	Software Engineering	3	1 -	3
5.	13A04507	Micro Processors & Interfacing	3	1 -	3
6.	13A52501	Managerial Economics and Financial Analysis	3	1 -	3
7.	13A05505	Operating Systems Lab	-	- 3	2
8.	13A05506	Compiler Design and Assembly Language Programming Lab	-	- 3	2
9.	13A52502	Advanced English language Comm. Skills Lab (Audit Course)	-	- 3	
Total Credits					22

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. III - I Sem.

Th	Tu	C
3	1	3

(13A05501) OPERATING SYSTEMS

Course Objective:

- To make the students understand the basic operating system concepts such as processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection.
- To get acquaintance with the class of abstractions afford by general purpose operating systems that aid the development of user applications.

Learning Outcome:

- Able to use operating systems effectively.
- Write System and application programs to exploit operating system functionality.
- Add functionality to the exiting operating systems
- Design new operating systems

UNIT I

Operating Systems Overview: Operating system functions, Operating system structure, operating systems Operations, protection and security, Kernel data Structures, Computing Environments, Open-Source Operating Systems

Operating System Structure: Operating System Services, User and Operating-System Interface, systems calls, Types of System Calls, system programs, operating system structure, operating system debugging, System Boot.

Processes: Process concept, process Scheduling, Operations on processes, Inter process Communication, Examples of IPC systems.

UNIT II

Threads: overview, Multicore Programming, Multithreading Models, Thread Libraries, Implicit threading, Threading Issues.

Process Synchronization: The critical-section problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic problems of synchronization, Monitors, Synchronization examples, Alternative approaches.

CPU Scheduling: Scheduling-Criteria, Scheduling Algorithms, Thread Scheduling, Multiple-Processor Scheduling, Real-Time CPU Scheduling, Algorithm Evaluation.

UNIT III

Memory Management: Swapping, contiguous memory allocation, segmentation, paging, structure of the page table.

Virtual memory: demand paging, page-replacement, Allocation of frames, Thrashing, Memory-Mapped Files, Allocating Kernel Memory

Deadlocks: System Model, deadlock characterization, Methods of handling Deadlocks, Deadlock prevention, Detection and Avoidance, Recovery from deadlock.

UNIT IV

Mass-storage structure: Overview of Mass-storage structure, Disk structure, Disk attachment, Disk scheduling, Swap-space management, RAID structure, Stable-storage implementation.

File system Interface: The concept of a file, Access Methods, Directory and Disk structure, File system mounting, File sharing, Protection.

File system Implementation: File-system structure, File-system Implementation, Directory Implementation, Allocation Methods, Free-Space management.

UNIT V

I/O systems: I/O Hardware, Application I/O interface, Kernel I/O subsystem, Transforming I/O requests to Hardware operations.

Protection: Goals of Protection, Principles of Protection, Domain of protection, Access Matrix, Implementation of Access Matrix, Access control, Revocation of Access Rights, Capability- Based systems, Language – Based Protection

Security: The Security problem, Program threats, System and Network threats, Cryptography as a security tool, User authentication, Implementing security defenses, Firewalling to protect systems and networks, Computer–security classifications.

Text Books:

1. *Operating System Concepts*, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, Ninth Edition, 2012, Wiley.
2. *Operating Systems: Internals and Design Principles*, Stallings, Sixth Edition, 2009, Pearson Education.

Reference Books:

1. *Modern Operating Systems*, Andrew S Tanenbaum, Second Edition, PHI.
2. *Operating Systems*, S.Haldar, A.A.Aravind, Pearson Education.
3. *Principles of Operating Systems*, B.L.Stuart, Cengage learning, India Edition.
4. *Operating Systems*, A.S.Godbole, Second Edition, TMH.
5. *An Introduction to Operating Systems*, P.C.P. Bhatt, PHI.
6. *Operating Systems*, G.Nutt, N.Chaki and S.Neogy, Third Edition, Pearson Education.
7. *Operating Systems*, R.Elmasri, A.G.Carrick and D.Levine, Mc Graw Hill.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. III-I Sem.

Th	Tu	C
3	1	3

(13A05502) COMPILER DESIGN

Course Objective:

The objectives of the course are

- *To realize the computer science as the basis for real time applications*
- *To introduce the major concept areas of language translation and compiler design.*
- *To learn how a compiler works and know about the powerful compiler generation tools and techniques, which are useful to the other non-compiler applications.*
- *To know the importance of code optimization.*

Learning Outcome:

Upon completion of this course, students will be:

- *Able to design a compiler for a simple programming language*
- *Able to use the tools related to compiler design effectively and efficiently*
- *Able to write the optimized code*

UNIT I

Introduction: Language processors, Phases of a compiler, Pass and phase, Bootstrapping, Compiler construction tools, Applications of compiler technology, Programming language basics

Lexical Analysis: Role and Responsibility, Input buffering, Specification of tokens, Recognition of tokens, LEX tool, Design of a Lexical Analyzer generator

UNIT II

Syntax Analysis: Role of the parser, Context Free Grammars - Definition, Derivations, Parse trees, Ambiguity, Eliminating ambiguity, Left recursion, Left factoring.

TOP Down Parsing: Recursive descent parsing, Non-recursive predictive parsing, LL(1) grammars, Error recovery in predictive parsing.

Bottom Up Parsing: Handle pruning, Shift-Reduce parsing, Conflicts during shifts- reduce parsing, SLR Parsing, Canonical LR(1) parsers, LALR parsers, Using ambiguous grammars, YACC tool.

UNIT III

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

Intermediated Code Generation: Need for intermediate code, Types of intermediate code, Three address code, Quadruples, Triples, Type expressions, Type equivalence, Type checking, Translation of expressions, control flow statements, switch statement, procedures, back patching.

UNIT IV

Run Time Storage Organization: Scope and Life time of variable, Information associated with symbols in symbol table, Data Structures for symbol Table, Static vs dynamic storage allocation, Stack allocation of space, Access to non-local data on stack, Heap management, Introduction to garbage collection

Optimization: Need and objective of optimization, Places of optimization, Optimization at user level, Construction of Basic blocks and Processing, Data Flow analysis using flow graph, Data flow equations for blocks with back ward flow control, Principles source of optimization and transformations, Alias, Loops in flow graphs, Procedural optimization, Loop optimization

UNIT V

Code Generation: Issues in code Generation, Target machine architecture, Subsequent Use information, Simple code generator, Register allocation, DAG representation of basic blocks, Code generation from intermediate code, Peephole optimization, Code scheduling

Text Books :

1. *Compilers Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman., Pearson.*
2. *Compiler Design, K. Muneeswaran., Oxford University Press, 2012*

Reference Books :

1. *Compiler Construction, K.V.N Sunitha, Pearson, 2013*
2. *Engineering a Compiler, Second Edition, Keith D. Cooper & Linda Torczon., Morgan Kaufmann, Elsevier.*
3. *Compilers Principles and Practice, Parag H. Dave, Himanshu B. Dave., Pearson*
4. *Compiler Design, Sandeep Saxena, Rajkumar Singh Rathore., S.Chand publications*
5. *Compiler Design, Santanu Chattopadhyay., PHI*
6. *Principals of Compiler Design, Nadhni Prasad, Elsevier.*

AMTUA

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. III - I Sem.

Th	Tu	C
3	1	3

(13A05503) UNIX AND SHELL PROGRAMMING

Course Objective:

This course is to provide a comprehensive introduction to Shell Programming.

Learning Outcome:

At the end of the course delegates will acquire

- *The fundamental skills required to write simple and complex Shell scripts to automate jobs and processes in the Unix environment.*

UNIT I

The UNIX Environment, Unix structure, Accessing UNIX, common and useful commands. The Vi Editor – Concepts, Modes and Commands. File Systems – File names and types, regular files and Directories and their implementation. Operations on directories, files and on both. Security levels, Changing permissions, Ownership and group

UNIT II

Shells- UNIX Session, standard streams, redirection, pipes tee Command, Command Execution and Substitution, Command-Line Editing, job control, Aliases, Variable Types and options, Shell Customization. Filters and Pipes – related Commands. Commands for Translating Characters, Files with duplicate Lines, Counting characters, words and Lines and Comparing files

UNIT III

User Communication, Electronic mail, Remote access, and File Transfer. Vi Editor – Local, Global and Range commands and Text manipulation in vi. Editor, and Over view of ex Editor. Atoms and Operators. grep – family and operations and searching for file contents. Overview of sed and awk

UNIT IV

Interactive korn shell and Korn shell Programming: An overview on sed. Korn shell - Features, Files, Variables, input and output. Environmental Variables and options. Startup Script, Command history and Execution process. Korn shell Programming- Script Concept, Expressions, Decision making and Repetition, Special Parameters and variables, Changing Positional parameters, Argument Validation, Debugging Scripts and Examples

UNIT V

Interactive C shell and C shell Programming: An overview on awk. C Shell – Features, Files and Variables, output, input, eval Command, environmental Variables, on-off Variables, Startup and Shutdown Scripts, Command history and execution Script. C Shell Programming – script Concepts, expressions, Decision making and repetition, Special Parameters, Changing Positional Parameters, argument Validation, Debugging Scripts and examples

Text Books :

1. *UNIX and Shell Programming, Behrouz A. Forouzan and Richard F. Gilberg, cengage learning publications, Indian Reprint 2012*
2. *Unix: The Ultimate Guide, Sumitabha Das, Tat Mcgraw-Hill Edition, Indian reprint 2012*

Reference Books :

1. *UNIX and Linux System Administration Handbook, Evi Nemeth, Garth Snyder, Trent R. Hein and Ben Whaley, PHI.*

2. *Essential Linux Administration: A Comprehensive Guide for Beginners*, Chuck Easttom, Cengage Learning
3. *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*, Michael Kerrisk, No Starch Press
4. *A Practical Guide to Linux Commands, Editors, and Shell Programming*, 3rd Edition, Mark G. Sobell, PHI
5. *Advanced Programming in the UNIX Environment*, 3rd Edition, W. Richard Stevens and Stephen A. Rago, Addison-Wesley professional
6. *UNIX Network Programming*, W. Richard Stevens, PHI

UNIVERSITY

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. III - I Sem.

Th	Tu	C
3	1	3

(13A05504) SOFTWARE ENGINEERING

Course Objective:

- To understand the software life cycle models.
- To understand the software requirements and SRS document.
- To understand the importance of modeling and modeling languages.
- To design and develop correct and robust software products.
- To understand the quality control and how to ensure good quality software.
- To understand the planning and estimation of software projects.
- To understand the implementation issues, validation and verification procedures.
- To understand the maintenance of software

Learning Outcome:

- Define and develop a software project from requirement gathering to implementation.
- Ability to code and test the software
- Ability to plan, Estimate and Maintain software systems

UNIT I

Software and Software Engineering: The Nature of Software, The Unique Nature of WebApps, Software Engineering, Software Process, Software Engineering Practice, Software Myths.

Process Models: A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Terminology, Product and Process.

UNIT II

Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Negotiating Requirements, Validating Requirements.

Requirements Modeling: Requirements Analysis, Scenario-Based Modeling, UML Models That Supplement the Use Case, Data Modeling Concepts, Class-Based Modeling.

UNIT III

Design Concepts: Design within the Context of Software Engineering, Design Process, Design Concepts, The Design Model.

Architectural Design: Software Architecture, Architectural Genres, Architectural Styles, Architectural Design.

Component-Level Design: What is a Component, Designing Class-Based Components, Conducting Component-Level Design, Component-Level Design for WebApps.

UNIT IV

User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Design Evaluation.

Coding and Testing: Coding, Code Review, Software Documentation, Testing, Testing in the Large versus Testing in the Small, Unit Testing, Black-Box Testing, White-Box Testing, Debugging, Program Analysis Tools, Integration Testing, Testing Object-Oriented Programs, System Testing, Some General Issues Associated with Testing.

UNIT V

Software Project Management: Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques,

COCOMO-A Heuristic Estimation Technique, Halstead's Software Science-An Analytical Technique, Staffing Level Estimation, Scheduling, Organization and Team Structures, Staffing, Risk Management, Software Configuration Management. .

Software Maintenance: Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Models, Estimation of Maintenance cost.

Text Books :

1. *Software Engineering A practitioner's Approach*, Roger S. Pressman, Seventh Edition, 2009, McGrawHill International Edition.
2. *Fundamentals of Software Engineering*, Rajib Mall, Third Edition, 2009, PHI.

Reference Books:

1. *Software Engineering*, Ian Sommerville, Ninth edition, Pearson education.
2. *Software Engineering : A Primer*, Waman S Jawadkar, Tata McGraw-Hill, 2008
3. *Software Engineering, A Precise Approach*, Pankaj Jalote, Wiley India, 2010.
4. *Software Engineering, Principles and Practices*, Deepak Jain, Oxford University Press.
5. *Software Engineering1: Abstraction and modeling*, Diner Bjorner, Springer International edition, 2006.
6. *Software Engineering2: Specification of systems and languages*, Diner Bjorner, Springer International edition , 2006.
7. *Software Engineering Foundations*, Yingxu Wang, Auerbach Publications, 2008.
8. *Software Engineering Principles and Practice*, Hans Van Vliet, 3rd edition, John Wiley & Sons Ltd.
9. *Software Engineering 3: Domains, Requirements, and Software Design*, D. Bjorner, Springer International Edition.
10. *Introduction to Software Engineering*, R.J. Leach, CRC Press.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. III - I Sem.

Th	Tu	C
3	1	3

(13A04507) MICROPROCESSORS & INTERFACING

Course Objective:

- Study the instruction set of 8086 microprocessor and its architecture
- Learn assembly language programming using 8086 microprocessor
- Interfacing 8051, 8255, 8237, and 8259

Learning Outcome:

- Program the 8086 microprocessor
- Interface the 8086 microprocessor with various devices and program them

UNIT I

Microprocessors-Evolution and Introduction: Microprocessors and Micro Controllers, Microprocessor based system, Origin of Microprocessor, Classification of Microprocessors, Types of Memory, I/O Devices, Technology Improvements Adapted to Microprocessors and Computers, Introduction to 8085 processor, Architecture of 8085, Microprocessor instructions, classification of instructions, Instruction set of 8085.

Intel 8086 Microprocessor architecture, Features, and Signals: Architecture of 8086, Accessing memory locations, PIN details of 8086.

UNIT II

Addressing Modes, Instruction Set and Programming of 8086: Addressing modes in 8086, Instruction set of 8086, 8086 Assembly Language Programming, Modular Programming.

UNIT III

8086 Interrupts: Interrupt types in 8086, Processing of Interrupts by 8086, Dedicated interrupt types in 8086, Software interrupts-types 00H-FFH, Priority among 8086 interrupts, Interrupt service routines, BIOS interrupts or functional calls, Interrupt handlers, DOS services-INT 21H, System calls-BIOS services.

Memory and I/O Interfacing: Physical memory organization in 8086, Formation of system bus, Interfacing RAM and EPROM chips using only logic gates, Interfacing RAM/ EPROM chips using decoder IC and logic gates, I/O interfacing, Interfacing 8-bit input device with 8086, Interfacing output device using 8086, Interfacing printer with 8086, Interfacing 8-bit and 16-bit I/O devices or ports with 8086, Interfacing CRT terminal with 8086.

UNIT IV

Features and Interfacing of programmable devices for 8086 systems: Intel 8255 programmable peripheral interface, Interfacing switches and LEDs, Interfacing seven segment displays, Traffic light control, Interfacing analog to digital converters, Intel Timer IC 8253, Introduction to serial communication, 8259 programmable controller, 8237 DMA controller.

UNIT V

Introduction to 8051 Micro controllers: Intel's MCS-51 series micro controllers, Intel 8051 architecture, Memory organization, Internal RAM structure, Power control in 8051, Stack operation.

8051 Instruction Set and Programming: Introduction, Addressing modes of 8051, Instruction set of 8051, Hardware features of 8051: Introduction, Parallel ports in 8051, External memory interfacing in 8051, Timers, Interrupts, Serial ports.

Interfacing Examples: Interfacing 8255 with 8051, Interfacing of push button switches and LEDs, Interfacing of seven segment displays.

Text Books:

1. *“Microprocessor and Interfacing 8086,8051, 8096 and advanced processors”*, Senthil Kumar, Saravanan, Jeevanathan, Shah, 1st edition, 2nd impression, 2012, Oxford University Press.
2. *“The X86 Microprocessors”*, Lyla B. Das. , 2010, Pearson.

Reference Books:

1. *“Microprocessor and Interfacing: Programming and Hardware”*, Douglas V.Hall, McGrawHill
2. *“8086 microprocessor: Programming and Interfacing the PC”*, Kenneth Ayala, Cengage Learning
3. *“ARM system-on-chip architecture”*, Steve Furber, Addison-Wesley Professional
4. *“The Intel Microprocessors”*, Barry B. Brey, Prentice Hall

AMTU

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. III-I Sem.

Th Tu C
3 1 3

(13A52501) MANAGERIAL ECONOMICS AND FINANCIAL ANALYSIS

Course Objective:

The objectives of this course are to equip the student with the basic inputs of Managerial Economics and Economic Environment of business and to enrich analytical skills in helping them take sound financial decisions for achieving higher productivity.

Learning Outcome:

The thorough understanding of Managerial Economics and Analysis of Financial Statements facilitates the Technocrats – cum – Entrepreneurs to take-up decisions effectively and efficiently in the challenging Business Environment.

UNIT I

INTRODUCTION TO MANAGERIAL ECONOMICS

Managerial Economics - Definition, nature and scope – contemporary importance of Managerial Economics - Demand Analysis: Determinants- Law of Demand - Elasticity of Demand. Significance – types – measurement of elasticity of demand - Demand forecasting- factors governing demand forecasting- methods of demand forecasting –Relationship of Managerial Economics with Financial Accounting and Management.

UNIT II

THEORY OF PRODUCTION AND COST ANALYSIS

Production Function – Short-run and long- run production - Isoquants and Isocosts, MRTS, least cost combination of inputs - Cobb-Douglas production function - laws of returns - Internal and External economies of scale - **Cost Analysis:** Cost concepts - Break-Even Analysis (BEA) - Managerial significance and limitations of BEA - Determination of Break Even Point (Simple Problems)

UNIT III

INTRODUCTION TO MARKETS AND NEW ECONOMIC ENVIRONMENT

Market structures: Types of Markets - Perfect and Imperfect Competition - Features, Oligopoly - Monopolistic competition. Price-Output determination - Pricing Methods and Strategies. Forms of Business Organization – Sole Proprietorship- Partnership – Joint Stock Companies – Public Sector Enterprises – New Economic Environment- Economic systems – Economic Liberalization – Privatization and Globalization

UNIT IV

CAPITAL AND CAPITAL BUDGETING

Concept of Capital - Over and Under capitalization – Remedial measures - Sources of Short term and Long term capital - Estimating Working Capital requirement – Capital budgeting – Features of Capital budgeting proposals – Methods and Evaluation of Capital budgeting – Pay Back Method – Accounting Rate of Return (ARR) – Net Present Value (NPV) – Internal Rate Return (IRR) Method (simple problems)

UNIT V

INTRODUCTION TO FINANCIAL ACCOUNTING AND ANALYSIS

Financial Accounting – Concept - emerging need and importance - Double-Entry Book Keeping- Journal - Ledger – Trial Balance - Financial Statements - - Trading Account – Profit & Loss Account – Balance Sheet (with simple adjustments). Financial Analysis – Ratios – Techniques – Liquidity, Leverage, Profitability, and Activity Ratios (simple problems).

Text Books:

1. Aryasri: *Managerial Economics and Financial Analysis*, 4/e, TMH, 2009.
2. Varshney & Maheswari: *Managerial Economics*, Sultan Chand, 2009.

Reference Books:

1. Premchand Babu, Madan Mohan: *Financial Accounting and Analysis*, Himalaya, 2009
2. S.A. Siddiqui and A.S. Siddiqui: *Managerial Economics and Financial Analysis*, New Age International, 2009.
3. Joseph G. Nellis and David Parker: *Principles of Business Economics*, Pearson, 2/e, New Delhi.
4. Domnick Salvatore: *Managerial Economics in a Global Economy*, Cengage, 2009.
5. H.L.Ahuja: *Managerial Economics*, S.Chand, 3/e, 2009

AMTUA

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. III - I Sem.

L C
3 2

(13A05505) OPERATING SYSTEM LAB

1. Practice session: practice use of some basic Linux commands. Document the syntax and semantics of those commands. Practice programs on shell variables, control statements etc.
2. Practice session: Study the features of Linux environment and submit a report on it.
3. Write a shell script that accepts a name from the user and displays whether it is a file, directory or something else.
4. Write a shell script that creates users
5. Write a shell script that searches for a given string in a file
6. Write a shell script that compiles all C files in your home directory and creates executable files
7. Write a shell script that given a filename as argument, deletes all even lines in a file
8. Implement the grep command in C language
9. Write a shell script that removes duplicate lines from a file
10. Write a shell script that enhances find command by adding error messages that explain why the command failed.
11. Write a shell script to backup files in a specified directory
12. Write a shell script that finds all links to a file
13. Write an awk script to count the number of lines in a file that do not contain vowels.
14. Write an awk script to find the number of characters, words and lines in a file.
15. Write C programs that illustrate communication between two unrelated processes using named pipe(FIFO File).
16. Write a C program in which a parent writes a message to a pipe and the child reads the message.
17. Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.
18. Write a C program (receiver.c) that receives the messages (from the above message queue and displays them.
19. Configure mail server and file server.
20. Write Client and Server programs in C for connection oriented communication between Server and Client processes using Unix Domain sockets to perform the following: Client process sends a message to the Server Process. The Server receives the message, reverses it and sends it back to the Client. The Client will then display the message to the standard output device.

Reference Books:

1. *Unix and Shell programming*, B.A.Forouzan and R.F.Gilberg, Cengage Learning.
2. *Beginning Linux Programming*, 4th Edition, N.Matthew, R.Stones,Wrox, Wiley
3. *Advanced Unix Programming*, N.B.Venkateswarulu, BS Publications.
4. *Unix and Shell Programming*, M.G. Venkatesh Murthy, Pearson Education.
5. *Unix Shells by Example*, 4th Edition, Ellie Quigley, Pearson Education.
6. *Sed and Awk*, O.Dougherty&A.Robbins, 2nd edition,SPD.
7. *Unix shell Programming*, S.G.Kochan and P.Wood, 3rd edition, Pearson Education.
8. *Shell Scripting*, S.Parker, Wiley India Pvt. Ltd.
9. *Advanced Programming in the Unix Environment*, 2nd edition, W.R.Stevens and S.A.Rago, Learson Education.
10. *Linux System Programming*, Robert Love, O'Reilly, SPD

PART-B

Simulate the following CPU scheduling algorithms

1. a) Round Robin b) SJF c) FCFS d) Priority
2. Simulate all file allocation strategies
a) Sequential b) Indexed c) Linked
3. Simulate MVT and MFT
4. Simulate all File Organization Techniques
a) Single level directory b) Two level c) Hierarchical d) DAG
5. Simulate Bankers Algorithm for Dead Lock Avoidance
6. Simulate Bankers Algorithm for Dead Lock Prevention
7. Simulate all page replacement algorithms
a) FIFO b) LRU c) LFU Etc. ...
8. Simulate Paging Technique of memory management

Reference Books :

1. *Operating System Concepts*, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, Eighth edition, John Wiley.
2. *Operating Systems: Internals and Design Principles*, Stallings, Sixth Edition–2009, Pearson Education
3. *Modern Operating Systems*, Andrew S Tanenbaum, Second Edition, PHI.
4. *Operating Systems*, S.Haldar, A.A.Aravind, Pearson Education.
5. *Principles of Operating Systems*, B.L.Stuart, Cengage learning, India Edition.
6. *Operating Systems*, A.S.Godbole, Second Edition, TMH.
7. *An Introduction to Operating Systems*, P.C.P. Bhatt, PHI.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR
III B.Tech. ISem.

L C
3 2

(13A05506) COMPILER DESIGN AND ASSEMBLY LANGUAGE PROGRAMMING
LAB

Compiler Design Lab

1. Write a program to search for a given pattern in a set of files. It should support regular expressions should work similar to grep and fgrep of Linux environment.
2. Write programs for DFA, NFA.
3. Consider the following regular expressions:
 - a) $(0 + 1)^* 1(0+1)(0+1)$
 - b) $(ab^*c + (def)^+ + a^*d^+e)^+$
 - c) $((a + b)^*(c + d)^+ + ab^*c^*d)$
 Write separate programs for recognizing the strings generated by each of the regular expressions mentioned above (Using FA).
4. Given a text-file which contains some regular expressions, with only one RE in each line of the file. Write a program which accepts a string from the user and reports which expression accepts that string. If no RE from the file accepts the string, then report that is matched.
5. Design a PDA for any given CNF. Simulate the processing of a string using the PDA and show parse tree.
6. Design a Lexical analyzer for identifying different types of tokens used in C language.
Note: The reserved keywords such as if, else, class, struct etc must be reported as invalid identifiers. C allows identifier names to begin with underscore character too.
7. Simulate a simple desktop calculator using any lexical analyzer generator tool (LEX or FLEX).
8. Program to recognize the identifiers, if and switch statements of C using a lexical analyzer generator tool.
9. Consider the following grammar:


```
S --> ABC
A--> abA | ab
B--> b | BC
C--> c | cC
```

 Design any shift reduced parser which accepts a string and tells whether the string is accepted by above grammar or not.
10. Design a YACC program that reads a C program from input file and identify all valid C identifiers and for loop statements.
11. Program to eliminate left recursion and left factoring from a given CFG.
12. YACC program that reads the input expression and convert it to post fix expression.
13. YACC program that finds C variable declarations in C source file and save them into the symbol table, which is organized using binary search tree.
14. YACC program that reads the C statements from an input file and converts them into quadruple three address intermediate code

Reference Books :

1. *Compiler Design using FLEX and YACC, Das, PHI.*
2. *Compiler Design in C, Holub, PHI.*

Assembly Language Programming Lab

1. Write an ALP to find factorial of number.
2. The 8 data bytes are stored from memory location E000H to E007H. Write 8086 ALP to transfer the block of data to new location B001H to B008H.
3. Write a program to display string Computer Science & Engineering for 8086.

4. Write a program to reverse the given string for 8086.
5. Write a program to multiply 2 numbers (16-bit data) for 8086.
6. Sum of series of 10 numbers and store result in memory location total.
7. Write a program to find Largest No. in a block of data. Length of block is 0A.
Store the maximum in location result.
8. Find number of times letter “e” exist in the string exercise, Store the count at memory
9. Write an assembly language program to count number of vowels in a given string.
10. Write an 8086 ALP which will input the user name from the keyboard. If the user is “Ramu-jntu” it will output “The username is valid” else it will output “Invalid user name”.

Reference Books :

1. *Microprocessor and Interfacing 8086,8051, 8096 and advanced processors*, Senthil Ku Saravanan, Jeevanathan, Shah, Oxford Publishers, 2012.
2. *8086 microprocessor: Programming and Interfacing the PC*, Kenneth Ayala, Cengage Learning
3. *The X86 Microprocessors*, Lyla B. Das. Pearson, 2012.

AMTU

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech. III - I Sem.

(13A52502) ADVANCED ENGLISH LANGUAGE COMMUNICATION SKILLS LAB (Audit Course)

Introduction:

The introduction of the Advanced Communication Skills Lab is considered essential at 3rd year level. At this stage, the students need to prepare themselves for their careers which may require them to listen to, read, speak and write in English both for their professional and interpersonal communication in the globalised context.

The proposed course should be a laboratory course to enable students to use 'good' English and perform the following:

- Gathering ideas and information to organise ideas relevantly and coherently.
- Engaging in debates.
- Participating in group discussions.
- Facing interviews.
- Writing project/research reports/technical reports.
- Making oral presentations.
- Writing formal letters.
- Transferring information from non-verbal to verbal texts and vice-versa.
- Taking part in social and professional communication.

Course Objective:

This Lab focuses on using multi-media instruction for language development to meet the following targets:

- To improve the students' fluency in English, through a well-developed vocabulary and enable them to listen to English spoken at normal conversational speed by educated English speakers and respond appropriately in different socio-cultural and professional contexts.
- Further, they would be required to communicate their ideas relevantly and coherently in writing.
- To prepare all the students for their placements.

Learning Outcome:

- Accomplishment of sound vocabulary and its proper use contextually
- Flair in Writing and felicity in written expression.
- Enhanced job prospects.
- Effective Speaking Abilities

The following course content to conduct the activities is prescribed for the Advanced English Language Communication Skills (AELCS) Lab:

UNIT I

COMMUNICATIVE COMPETENCY

1. Reading Comprehension
2. Listening comprehension
3. Vocabulary for competitive purpose
4. Spotting errors

UNIT II

TECHNICAL WRITING

1. Report writing
2. Curriculum vitae
3. Covering letter
4. E-mail writing

UNIT III

PRESENTATIONAL SKILLS

1. Oral presentation
2. Power point presentation
3. Poster presentation
4. Stage dynamics

UNIT IV

CORPORATE SKILLS

1. Dress code
2. Telephonic skills
3. Net Etiquettes

UNIT V

GETTING READY FOR JOB

1. Group discussions
2. Interview skills
3. Psychometric tests

Minimum Requirement:

The Advanced English Language Communication Skills (AELCS) Laboratory shall have the following infra-structural facilities to accommodate at least 60 students in the lab:

- Spacious room with appropriate acoustics.
- Round Tables with movable chairs
- Audio-visual aids
- LCD Projector
- Public Address system
- P – IV Processor, Hard Disk – 80 GB, RAM–512 MB Minimum, Speed – 2.8 GHZ
- T. V, a digital stereo & Camcorder
- Headphones of High quality

Suggested Software:

The software consisting of the prescribed topics elaborated above should be procured and used.

K-VAN SOLUTIONS-Advanced communication lab

1. DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.
2. TOEFL & GRE(KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)
3. Train2success.com

References:

1. Objective English For Competitive Exams, Hari Mohana Prasad, 4th edition, Tata Mc Graw Hill.
2. Technical Communication by Meenakshi Raman & Sangeeta Sharma, O U Press 2009.
3. Books on TOEFL/GRE/GMAT/CAT/ IELTS by Barron's/DELTA/Cambridge University Press.2012.
4. Soft Skills for Everyone, Butterfield Jeff, Cengage Publications, 2011.
5. Practice Psychometric Tests: How to familiarize yourself with genuine recruitment tests, 2012.
6. Management Shapers Series by Universities Press (India) Pvt Ltd., Himayatnagar, Hyderabad 2008.
7. Handbook for Technical Writing by David A McMurrey & Joanne Buckely CENGAGE Learning 2008.
8. English for Technical Communication for Engineering Students, Aysha Vishwamohan, Tata Mc Graw-Hill 2009.
9. Word Power Made Handy, Shalini Verma, S Chand Publications, 2011.
10. Effective Technical Communication, Ashrif Rizvi, TataMcGrahill, 2011.