Program Structure B.E. Information Technology, (Rev. 2016)

S. E. Information Technology (Semester-IV)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract	Tut	Theory	TW/ Pract	Tut	Total
ITC401	Applied Mathematics-IV	4+1@	-	-	5	-	-	5
ITC402	Computer Networks	4	-	-	4	-	-	4
ITC403	Operating Systems	4	-	-	4	-	-	4
ITC404	Computer Organization and Architecture	4	-	-	4	-	-	4
ITC405	Automata Theory	3+1$	-	-	4	-	-	4
ITL401	Networking Lab	-	2	-	-	1	-	1
ITL402	Unix Lab	-	2	-	-	1	-	1
ITL403	Microprocessor Programming Lab	-	2	-	-	1	-	1
ITL404	Python Lab	-	2+2*	-	-	2	-	2
	Total	**21**	**10**	-	**21**	**5**	-	**26**

ourse Code	Course Name	Examination Scheme								
		Theory					TW	Oral	Oral & Pract	Total
		Internal Assessment			End Sem. Exam	Exam Duration (in Hrs)				
		Test 1	Test 2	Avg.						
ITC401	Applied Mathematics-IV	20	20	20	80	3	-	-	-	100
ITC402	Computer Networks	20	20	20	80	3	-	-	-	100
ITC403	Operating Systems	20	20	20	80	3	-	-	-	100
ITC404	Computer Organization and Architecture	20	20	20	80	3	-	-	-	100
ITC405	Automata Theory	20	20	20	80	3	--	-	-	100
ITL401	Networking Lab	-	-	-	-	-	25	25	--	50
ITL402	Unix Lab	-	-	-	-	-	25	--	25	50
ITL403	Microprocessor Programming Lab	-	-	-	-	-	25	25	--	50
ITL404	Python Lab	-	-	-	-	-	50	--	50	100
	Total	100	100	100	400	-	125	50	75	750

@ 4 hours shown as theory to be taken class wise and 1 hour to be taken tutorial as class wise

$ 3 hours shown as theory to be taken class wise and 1 hour to be taken tutorial as batch wise

*2 hours shown as practical's to be taken class wise lecture and other 2 hours to be taken as batch wise practicals in Lab.

Course Code	Course Name	Theory	Practical	Tutorial	Theory	Oral & Practical	Tutorial	Total
ITC401	Applied Mathematics IV	04	--	01	04	--	--	05

Course Code	Course Name	Examination Scheme							
		Theory Marks				Term Work	Oral & Practical	Oral	Total
		Internal assessment			End Sem. Exam				
		Test1	Test2	Avg. of Two Tests					
ITC401	Applied Mathematics IV	20	20	20	80	--	--	--	100

Course Objectives: Students will try to learn:

1. The concepts of Number Theory by using different theorem.
2. The concepts of probability and study PDF.
3. The concept of sampling theory and correlation.
4. The concept of graphs and trees.
5. The concept of groups theory.
6. The concept of Lattice theory.

Course Outcomes: Students will able to:

1. Apply the Number Theory to different applications using theorem.
2. Apply probability and understand PDF.
3. Understand sampling theory and correlation.
4. Apply the graphs and trees concepts to different applications.
5. Understand group's theory.
6. Understand the Lattice theory.

Prerequisite: Applied Mathematics III

Detailed syllabus:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Basic of Set, Permutations, Combination and Probability .	02	
I	Elements of Number Theory I	Modular Arithmetic, Divisibility and Euclid Algorithm, Primes and the Sieve of Eratosthenes, Testing for primes, Prime Number Theorem	06	CO1

II	Elements of Number Theory II	Euler's, Fermat's Little theorems, Congruences, Computing Inverse in Congruences, Legendre and Jacobi Symbols, Chinese Remainder Theorem	06	CO1
III	Probability	Statistics: Formal concept, sample space, outcomes, events Random Variables: discrete & continuous random variables, expectation, Variance, Probability Density Function & Cumulative Density Function Moments, Moment Generating Function Probability distribution: binomial distribution, Poisson & normal distribution	08	CO2
IV	Sampling theory	Test of Hypothesis, Level of significance, Critical region, One Tailed and two Tailed test, Test of significant for Large Samples:- Means of the samples and test of significant of means of two large samples Test of significant of small samples:- Students t- distribution for dependent and independent samples Chi square test:- Test of goodness of fit and independence of attributes, Contingency table. Correlation Scattered diagrams Karl Pearson's coefficient of correlation Spearman's Rank correlation Regression Lines	10	CO3
V	Graph & Groups theory.	Introduction to graphs, graph terminology, representing graphs and graph isomorphism, connectivity, Euler and Hamilton paths, planar graphs, graph coloring, introduction to trees, application of trees. Groups, subgroups, generators and evaluation of powers, cosets and Lagrange's theorem, permutation groups and Burnside's theorem, isomorphism, automorphisms, homomorphism and normal	12	CO4 CO5

		subgroups, rings, integral domains and fields.		
VI	Lattice theory	Lattices and algebras systems, principles of duality, basic properties of algebraic systems defined by lattices, distributive and complimented lattices, Boolean lattices and Boolean algebras, uniqueness of finite Boolean expressions, prepositional calculus. Coding theory: Coding of binary information and error detection, decoding and error correction.	08	CO5

Text Books:

1. Cryptograph and Network Security by B. A. Forouzan & D. Mukhopadhyay, 11th edition, McGraw Hill Publication.
2. Network Security and Cryptograph byBernard Menezes, Cengage Learning Publication.
3. Higher Engineering Mathematics by Grewal B. S. 38th edition, Khanna Publication 2005.
4. Probability and Statistics for Engineering, Dr. J Ravichandran, Wiley-India.
5. Mathematical Statistics by H. C Saxena, S Chand & Co.
6. *C. L. Liu: Elements of Discrete Mathematics*, 2nd edition, TMH

References:

1. Elementary Number Theory and its applications by Kenneth H. Rosen, 5th edition, Addison Wesley Publication.
2. Abstract Algebra by I. N. Herstain, 3rd eition, John Wiley and Sons Publication.
3. Discrete Mathematics by Norman Biggs, 2nd edition, Oxford University Press.
4. Advanced Engg. Mathematics by C. Ray Wylie & Louis Barrett.TMH International Edition.
5. Mathematical Methods of Science and Engineering by Kanti B. Datta, Cengage Learning.
6. Advanced Engineering Mathematics by Kreyszig E. 9th edition, John Wiley.
7. Probability by Seymour Lipschutz, McGraw-Hill publication.

Assessment:

Internal Assessment for 20 marks:
Consisting of **Two Compulsory Class Tests**
Approximately 40% to 50% of syllabus content must be covered in First test and remaining 40% to 50% of syllabus contents must be covered in second test.

End Semester Examination: Some guidelines for setting the question papers are as:
- Weightage of each module in end semester examination is expected to be/will be proportional to number of respective lecture hours mentioned in the syllabus.
- Question paper will comprise of total **six questions**, **each carrying 20 marks.**
- **Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**.

- **Remaining question will be mixed in nature** (for example if Q.2 has part (a) from module 3 then part (b) will be from any other module. (Randomly selected from all the modules.)
- Total **four questions** need to be solved.

Course Code	Course Name	Theory	Practical	Tutorial	Theory	Oral & Practical	Tutorial	Total
ITC402	Computer Networks	04	--	--	04	--	--	04

Course Code	Course Name	Examination Scheme							
		Theory Marks				Term Work	Oral & Practical	Oral	Total
		Internal assessment			End Sem. Exam				
		Test1	Test 2	Avg. of two Tests					
ITC402	Computer Networks	20	20	20	80	--	--	--	100

Course Objectives: Students will try to:

1. Study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.
2. Acquire knowledge of Application layer and Presentation layer paradigms and protocols.
3. Study Session layer design issues, Transport layer services, and protocols.
4. Gain core knowledge of Network layer routing protocols and IP addressing.
5. Study data link layer concepts, design issues, and protocols.
6. Read the fundamentals and basics of Physical layer, and will apply them in real time applications.

Course Outcomes: Students will be able to:

1. Describe the functions of each layer in OSI and TCP/IP model.
2. Explain the functions of Application layer and Presentation layer paradigms and Protocols.
3. Describe the Session layer design issues and Transport layer services.
4. Classify the routing protocols and analyze how to assign the IP addresses for the given network.
5. Describe the functions of data link layer and explain the protocols.
6. Explain the types of transmission media with real time applications.

Prerequisite: COA, Logic Design

Detailed syllabus:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Von Neumann model, Modulation, Demodulation, encoding, Decoding.	02	--

I	Introduction	Network Criteria, Physical Structures, Network Types: LAN, WAN, Switching, OSI Reference model, TCP/IP suite, Comparison of OSI and TCP/IP, Network devices.	04	CO1
II	Application layer and Presentation layer	Introduction: Providing Services, Application layer Paradigms, Client-Server Paradigm: Application Programming Interface, Using Services of the Transport Layer, Standard Client Server applications: World Wide Web and HTTP, FTP, Electronic Mail, TELNET, Secure Shell (SSH), Domain Name System (DNS), Compression: Lossless Compression, Lossy Compression, Multimedia data: Text, Image, Video , Audio ,Multimedia in the Internet: Streaming Stored Audio/Video, Streaming Live Audio/Video, Real-Time Interactive Audio/Video, Optimal Compression Algorithms, Huffman Coding, Adaptive Huffman Compression, Dictionary Based Compression, Speech Compression, LZW, RLE, Image Compression – GIF,JPEG.	10	CO1 CO2
III	Session layer and Transport layer	Session layer design issues, Session Layer protocol - Remote Procedure Call (RPC), Transport layer services, Transport Layer Protocols: Simple Protocol, Stop-and-Wait Protocol, Go-Back-N Protocol (GBN), Selective-Repeat Protocol, Bidirectional Protocols: Piggybacking, Internet Transport-Layer Protocols, User Datagram Protocol: User Datagram, UDP Services, UDP Applications, Transmission Control Protocol: TCP Services, TCP Features, Segment, Segment, A TCP Connection, State Transition Diagram, Windows in TCP, Flow Control, Error Control, TCP Congestion Control, TCP Timers, Options.	10	CO1 CO3
IV	Network Layer	Introduction: Network-Layer Services, Packet Switching, Network-Layer Performance, Network-Layer Performance, Network-Layer Congestion, Structure of A Router, Network Layer Protocols: IPv4 Datagram Format, IPv4 Addresses,	12	CO1 CO4

		Forwarding of IP Packets, ICMPv4, Unicast Routing: General Idea, Routing Algorithms, Unicast Routing Protocols, Multicast Routing : Introduction, Multicasting Basics, Intradomain Routing Protocols, Interdomain Routing Protocols, Next generation IP: Packet Format , IPv6 Addressing , Transition from IPv4 to IPv6, ICMPv6, Mobile IP: Addressing , Agents , Three Phases , Inefficiency in Mobile IP.		
V	Data Link Layer	Wired Networks; Introduction: Nodes and Links, Two Types of Links, Two Sublayers, Data Link Control: Framing, Flow and Error Control, Error Detection and Correction, Two DLC Protocols, Medium Access Protocols: Random Access, Controlled Access, Channelization, Link Layer Addressing, Wired LANS: Ethernet Protocol; IEEE Project 802, Standard Ethernet, Fast Ethernet (100 Mbps), Gigabit Ethernet, 10-Gigabit Ethernet, Virtual LANs, Other Wired Networks: Point-to-Point Networks, SONET, Switched Network: ATM, Connecting Devices: Repeaters or Hubs, Link-Layer Switches, Routers, Sliding Window Compression.	09	CO1 CO5
VI	Physical Layer	Data and Signals: Analog and Digital, Transmission Impairment, Data Rate Limits, Performance, Digital Transmission: Digital-to-Digital Conversion , Analog-to-Digital Conversion, Analog Transmission: Digital-to-Analog Conversion, Analog-to-Analog Conversion ,Bandwidth Utilization: Multiplexing, Spread Spectrum, Transmission Media: Guided Media, Unguided Media: Wireless, Real Time Interactive Protocols: Rationale for New Protocols, RTP, Session Initialization Protocol (SIP), H.323, SCTP.	05	CO1 CO6

Text Books:

1. Behrouz A. Forouzan, Forouzan Mosharrat , Computer Networks A Top down Approach, Mc Graw Hill education.
2. Andrew S Tanenbaum, Computer Networks -, 4th Edition, Pearson Education.
3. Ranjan Bose, Information Theory, Coding and Cryptography, Ranjan Bose, Tata McGrawHill , Second Edition.

4. Diane Teare, " Authorized Self- Study Guide Designing for CISCO Internetwork Solutions(DESGN), Second Edition.

References:

1. Behrouz A. Forouzan, Data communications and Networking, Fifth edition TMH 2013.
2. James F. Kurose, K. W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 3rd Edition, Pearson Eduction.
3. L. L. Peterson and B. S. Davie, Computer Networks: A Systems Approach, 4th Ed, Elsevier India.
4. S. Keshav, An Engineering Approach to Computer Networks, 2nd Edition, Pearson Edication.
5. W. A. Shay, Understanding communications and Networks, 3rd Edition, W. A. Shay, Cengage Learning.
6. Khalid Sayood, Introduction to Data Compression, Third Edition, Morgan Kaufman.

Assessment:

Internal Assessment for 20 marks:
Consisting of **Two Compulsory Class Tests**
Approximately 40% to 50% of syllabus content must be covered in First test and remaining 40% to 50% of syllabus contents must be covered in second test.

End Semester Examination: Some guidelines for setting the question papers are as:
- Weightage of each module in end semester examination is expected to be/will be proportional to number of respective lecture hours mentioned in the syllabus.
- Question paper will comprise of total **six questions**, **each carrying 20 marks.**
- **Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**.
- **Remaining question will be mixed in nature** (for example if Q.2 has part (a) from module 3 then part (b) will be from any other module. (Randomly selected from all the modules.)
- Total **four questions** need to be solved.

Course Code	Course Name	Theory	Practical	Tutorial	Theory	Oral & Practical	Tutorial	Total
ITC403	Operating System	04	--	--	04	--	--	04

Course Code	Course Name	Examination Scheme							
		Theory Marks				Term Work	Oral & Practical	Oral	Total
		Internal assessment			End Sem. Exam				
		Test1	Test 2	Avg. of two Tests					
ITC403	Operating System	20	20	20	80	--	--	--	100

Course Objectives: Students will try:

1. To understand the main components of an OS & their functions.
2. To study the process management and scheduling.
3. To understand various issues in Inter Process Communication (IPC) and the role of OS in IPC.
4. To understand the concepts and implementation Memory management policies and virtual memory.
5. To understand the working of an OS as a resource manager, file system manager, process manager, memory manager and I/O manager and methods used to implement the different parts of OS
6. To study the need for special purpose operating system with the advent of new emerging technologies

Course Outcomes: Student will be able to

1. Describe the important computer system resources and the role of operating system in their management policies and algorithms.
2. Understand the process management policies and scheduling of processes by CPU
3. Evaluate the requirement for process synchronization and coordination handled by operating system
4. Describe and analyze the memory management and its allocation policies.
5. Identify use and evaluate the storage management policies with respect to different storage management technologies.
6. Identify the need to create the special purpose operating system.

Prerequisite: Programming Language C

Detailed syllabus:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Programming Language C. Basic of Hardware i.e. ALU,RAM,ROM, HDD etc.	02	
I	Overview of Operating System	Introduction: Operating System Structure and operations, Process management, Memory management, storage management, Protection and security, Distributed and special purpose Systems; System Structure: Operating system services and interface, System calls and its types, System programs, Operating System Design and implementation, OS structure, Virtual machines, OS debugging and generation, System boot.	07	C01
II	Process Management	Process concept: Process Scheduling, Operation on process and Interprocess communication;, Multithreading, Process: Multithreading models and thread libraries, threading issues; Process Scheduling: Basic concepts, Scheduling algorithms and Criteria, Thread Scheduling and Multiple Processor Scheduling;	09	C02
III	Process coordination	Synchronization: The critical Section Problem, Peterson's Solution, synchronization Hardware and semaphores, Classic problems of synchronization, monitors, Atomic transactions; Deadlocks: System Model, Deadlock Characterization , Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance , Deadlock Detection, Recovery from Deadlock.	09	CO3
IV	Memory Management	Memory Management strategies: Background, Swapping, Contiguous Memory Allocation, Paging , Structure of the Page Table, Segmentation; Virtual Memory Management: Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files, Allocating Kernel Memory, Other Considerations.	10	C04
V	Storage Management	File system: File Concept , Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection; Implementing file System: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, NFS; Secondary Storage Structure: Overview of Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, RAID Structure, Stable-Storage Implementation, Tertiary-Storage Structure, Swap-Space Management; I/O systems: Overview I/0 Hardware, Application I/0 Interface, Kernel I/0 Subsystem, Transforming I/0 Requests to	09	C05

		Hardware Operations, STREAMS, Performance		
VI	Distributed Systems	Distributed operating System: Network based OS, Network Structure and Topology, Communication Structure and Protocols; Distributed File system: Naming and transparency, Remote file access, Stateful Versus Stateless Service, File Replication; Distributed Synchronization: Mutual Exclusion, Concurrency Control and Deadlock Handling,	06	C06

Text Books:

1. Operating System Concepts, Abraham Silberschatz, Greg Gagne, Peter Baer Galvin, 8th edition Wiley.
2. Modern Operating System, Tanenbaum, Pearson Education.
3. Operating Systems: Internal and Design Principles: William Stallings, PHI

Reference Books:

1. Operating System Design and Implementation, A Tanenbaum, Pearson
2. Real Time Systems Design and Analysis, Wiley, IEEE Press
3. Principles of Operating Systems: Naresh Chauhan, Oxford Higher Education

Assessment:

Internal Assessment for 20 marks:
Consisting of **Two Compulsory Class Tests**
Approximately 40% to 50% of syllabus content must be covered in First test and remaining 40% to 50% of syllabus contents must be covered in second test.

End Semester Examination: Some guidelines for setting the question papers are as:
- Weightage of each module in end semester examination is expected to be/will be proportional to number of respective lecture hours mentioned in the syllabus.
- Question paper will comprise of total **six questions**, **each carrying 20 marks.**
- **Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**.
- **Remaining question will be mixed in nature** (for example if Q.2 has part (a) from module 3 then part (b) will be from any other module. (Randomly selected from all the modules.)
- Total **four questions** need to be solved.

Course Code	Course Name	Theory	Practical	Tutorial	Theory	Oral & Practical	Tutorial	Total
ITC404	Computer Organization and Architecture	04	--	--	04	--	--	04

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Oral & Practical	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg. of two Tests				
ITC404	Computer Organization and Architecture	20	20	20	80	--	--	100

Course Objectives: Students will try to:

1. Conceptualize the basics of organizational and architectural issues of a digital computer.
2. Analyze processor performance improvement using instruction level parallelism.
3. Learn the function of each element of a memory hierarchy.
4. Study various data transfer techniques in digital computer.
5. Articulate design issues in the development of processor or other components that satisfy design requirements and objectives.
6. Learn microprocessor architecture and study assembly language programming.

Course Outcomes: Students will be able to:

1. Describe basic organization of computer and the architecture of 8086 microprocessor.
2. Implement assembly language program for given task for 8086 microprocessor.
3. Demonstrate control unit operations and conceptualize instruction level parallelism.
4. Demonstrate and perform computer arithmetic operations on integer and real numbers.
5. Categorize memory organization and explain the function of each element of a memory hierarchy.
6. Identify and compare different methods for computer I/O mechanisms.

Prerequisite: Fundamentals of Computer, Digital Logic Design

Detailed syllabus:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	basic combinational and sequential logic circuits, binary numbers and arithmetic, basic computer organizations	02	
I	Overview of Computer Architecture &	Introduction of Computer Organization and Architecture. Basic organization of computer and block level description of the functional	07	CO1

	Organization	units. Evolution of Computers, Von Neumann model. Performance measure of Computer Architecture. Architecture of 8086 family, 8086 Hardware Design, Minimum mode & Maximum mode of Operation. Study of bus controller 8288 & its use in Maximum mode.		
II	Programming 8086	Addressing modes, Instruction Set, Assembly Language Programming, Mixed Language Programming, Programs based on Stacks, Strings, Procedures, Macros, Timers, Counters & delay.	10	CO2
III	Processor Organization and Architecture	CPU Architecture, Register Organization, Instruction formats, basic instruction cycle. Instruction interpretation and sequencing. Control Unit: Soft wired (Micro-programmed) and hardwired control unit design methods. Microinstruction sequencing and execution. Micro operations, concepts of nano programming. Introduction to parallel processing concepts, Flynn's classifications, pipeline processing, instruction pipelining, pipeline stages, pipeline hazards.	11	CO3
IV	Data Representation and Arithmetic Algorithms	Number representation: Binary Data representation, two's complement representation and Floating-point representation. Integer Data arithmetic: Addition, Subtraction. Multiplication: Unsigned & Signed multiplication- Add & Shift Method, Booth's algorithm. Division of integers: Restoring and non-restoring division, signed division, basics of floating point representation IEEE 754 floating point(Single & double precision) number representation. Floating point arithmetic: Addition, subtraction	10	CO4
V	Memory Organization	Introduction to Memory and Memory parameters. Classifications of primary and secondary memories. Types of RAM and ROM, Allocation policies, Memory hierarchy and characteristics. Cache memory: Concept, architecture (L1, L2, L3), mapping techniques. Cache Coherency, Interleaved and Associative memory.	07	CO5
VI	I/O Organization	Input/output systems, I/O modules and 8089 IO processor. Types of data transfer techniques: Programmed I/O, Interrupt driven I/O and DMA.	05	CO6

Text Books:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", Fifth Edition, Tata McGraw-Hill.
2. William Stallings, "Computer Organization and Architecture: Designing for Performance", Eighth Edition, Pearson.
3. 8086/8088 family: Design Programming and Interfacing: By John Uffenbeck (Pearson Education)
4. Microprocessor and Interfacing: By Douglas Hall (TMH Publication).

References:

1. B. Govindarajulu, "Computer Architecture and Organization: Design Principles and Applications", Second Edition, Tata McGraw-Hill.
2. Dr. M. Usha, T. S. Srikanth, "Computer System Architecture and Organization", First Edition, Wiley-India.
3. John P. Hayes, "Computer Architecture and Organization", McGraw-Hill.,Third Edition.
4. K Bhurchandi, "Advanced Microprocessors & Peripherals", Tata McGraw-Hill Education

Assessment:

Internal Assessment for 20 marks:
Consisting of **Two Compulsory Class Tests**
Approximately 40% to 50% of syllabus content must be covered in First test and remaining 40% to 50% of syllabus contents must be covered in second test.

End Semester Examination: Some guidelines for setting the question papers are as:

- Weightage of each module in end semester examination is expected to be/will be proportional to number of respective lecture hours mentioned in the syllabus.
- Question paper will comprise of total **six questions**, **each carrying 20 marks.**
- **Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**.
- **Remaining question will be mixed in nature** (for example if Q.2 has part (a) from module 3 then part (b) will be from any other module. (Randomly selected from all the modules.)
- Total **four questions** need to be solved.

Course Code	Course Name	Theory	Practical	Tutorial	Theory	Oral & Practical	Tutorial	Total
ITC405	Automata Theory	03	--	01	03	--	01	04

Course Code	Course Name	Examination Scheme							
		Theory Marks				Term Work	Oral & Practical	Oral	Total
		Internal assessment			End Sem. Exam				
		Test1	Test 2	Avg. of two Tests					
ITC405	Automata Theory	20	20	20	80	--	--	--	100

$ 3 hours shown as theory to be taken class wise and 1 hour to be taken tutorial as batch wise

Course Objectives: Students will try:

1. To learn fundamentals of Regular and Context Free Grammars and Languages
2. To understand the relation between Regular Language and Finite Automata and machines.
3. To learn how to design Automata's and machines as Acceptors, Verifiers and Translators.
4. To understand the relation between Contexts free Languages, PDA and TM.
5. To learn how to design PDA as acceptor and TM as Calculators.
6. To learn how to co-relate Automata's with Programs and Functions.

Course Outcomes: The students will be able to:

1. Understand, design, construct, analyze and interpret Regular languages, Expression and Grammars.
2. Design different types of Finite Automata and Machines as Acceptor, Verifier and Translator.
3. Understand, design, analyze and interpret Context Free languages, Expression and Grammars.
4. Design different types of Push down Automata as Simple Parser.
5. Design different types of Turing Machines as Acceptor, Verifier, Translator and Basic computing machine.
6. Compare, understand and analyze different languages, grammars, Automata and Machines and appreciate their power and convert Automata to Programs and Functions

Prerequisite: Basic Mathematical Fundamentals: Sets, Logic, Relations, Functions.

Detailed syllabus:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
I	Introduction and Regular	Languages: Alphabets and Strings. Regular Languages: Regular Expressions, Regular Languages, Regular Grammars, RL and LL	06	CO1

	Languages	grammars, Closure properties		
II	Finite Automata and machines	Finite Automata: FA as language acceptor or verifier, NFA (with and without ε) , DFA, RE to NFA, NFA to DFA, Reduced DFA , NFA-DFA equivalence, FA to RE. Finite State Machines: m/c with output Moore and Mealy machines. M/c as translators. Melay and Moore m/c conversion	09	CO2
III	Context Free Grammars	Context Free Languages: CFG, Leftmost and Rightmost derivations, Ambiguity, Simplification and Normalization (CNF) and Chomskey Hierarchy (Types 0 to 3)	08	CO3
IV	Push Down Automata	Push Down Automata: Deterministic (single stack)PDA, Equivalence between PDA and CFG.	05	CO4
V	Turing Machine	Turing Machine: Deterministic TM , Multi-track and Multi-tape TMs, concept of UTM and idea of system program. Issue and concept of Halting Problem	07	CO5
VI	Applications of Automata	1.Power and Limitations of Regular and Context Free Grammars and Machines 2.Designing Functions: FA: Acceptor and Verifier. FSM: Translator PDA: Simple Parser for WF parenthesis, palindromes etc. TM: Basic bit wise calculator(+ /-/AND/OR) and Translator (**Note Added)**	04	CO2 CO4 CO5 CO6

Text books

1. J.C.Martin, "Introduction to languages and the Theory of Computation", TMH.
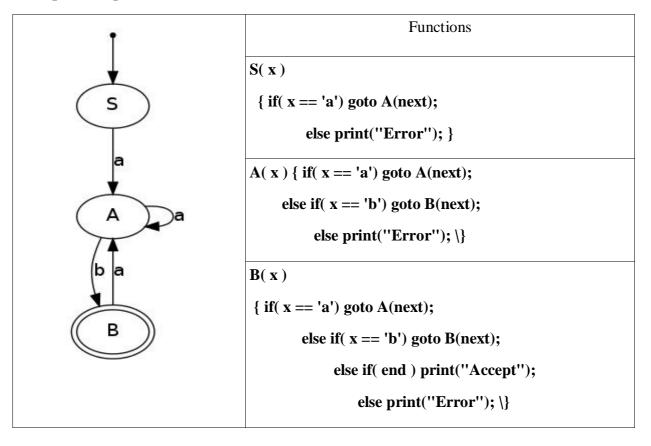2. Kavi Mahesh, "Theory of Computation A Problem Solving Approach", Wiley India

References

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata Theory, Languages and Computation", Pearson Education.
2. Daniel I.A. Cohen, "Introduction to Computer Theory", John Wiley & Sons.
3. Theory of Computation - By Vivek Kulkarni from Oxford University.
4. N.Chandrashekhar& K.L.P. Mishra, "Theory of Computer Science, Automata Languages & Computations", PHI publications.

Sample Example for Tutorial: Applications of Automata

An automata can be easily converted to functions by converting *States* to *functions* and *Transitions* to *function calls* or *gotos* begining with Starting state and *Accepting* in a terminating state.

A simple example of DFA is:

	Functions
	S(x) { if(x == 'a') goto A(next); else print("Error"); }
	A(x) { if(x == 'a') goto A(next); else if(x == 'b') goto B(next); else print("Error"); \}

	B(x)
	
	 { if(x == 'a') goto A(next);
	
	 else if(x == 'b') goto B(next);
	
	 else if(end) print("Accept");
	
	 else print("Error"); \}

Suggested Tutorials:

Sr. No.	Module	Detailed Content
I	Introduction and Regular Languages	1 Tutorial on design of RE, RG, RLG and LLG for given Regular Language.
II	Finite Automata and machines	3 Tutorials for converting RE to NFA, NFA to DFA to Reduced DFA, FA to RE. 1 Tutorial on design of Moore and Mealy machines.
III	Context Free Grammars	1 Tutorial on design of CFG and Leftmost and Rightmost derivations. 1 Tutorial for converting CFG to CNF.
IV	Push Down Automata	1 Tutorial on design of Push Down Automata.
V	Turing Machine	1 Tutorial on design of single tape Turing Machine. 1 Tutorial on design of Multi-track and Multi-tape TMs.
VI	Applications of Automata	2 Tutorials for converting Automata to Functions: a. FA to Acceptor / Verifier. b. FSM to Translator. c. PDA to Simple Parser for WF parenthesis, palindromes etc. d. TM to Basic bit wise calculator(+ /- /AND/OR) / Translator

Assessment:

Internal Assessment for 20 marks:
Consisting of **Two Compulsory Class Tests**
Approximately 40% to 50% of syllabus content must be covered in First test and remaining 40% to 50% of syllabus contents must be covered in second test.

End Semester Examination: Some guidelines for setting the question papers are as:
- Weightage of each module in end semester examination is expected to be/will be proportional to number of respective lecture hours mentioned in the syllabus.
- Question paper will comprise of total **six questions**, **each carrying 20 marks.**
- **Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**.
- **Remaining question will be mixed in nature** (for example if Q.2 has part (a) from module 3 then part (b) will be from any other module. (Randomly selected from all the modules.)
- Total **four questions** need to be solved.

Course Code	Course Name	Theory	Practical	Tutorial	Theory	TW/Prac tical	Tutorial	Total
ITL401	Networking Lab	--	02	--	--	1	--	1

Course Code	Course Name	Examination Scheme				Term Work	Oral	Total
		Theory Marks						
		Internal assessment			End Sem. Exam			
		Test 1	Test 2	Avg. of 2 Tests				
ITL401	Networking Lab	--	--	--	--	25	25	50

Lab Objectives: Students will try:

1. To get familiar with the basic network administration commands.
2. To install and configure network simulator and learn basics of TCL scripting.
3. To understand the network simulator environment and visualize a network topology and observe its performance
4. To analyze the traffic flow and the contents of protocol frames.
5. To implement client-server socket programs.
6. To design and configure a network for an organization.

Lab Outcomes: Student will be able to

1. Execute and evaluate network administration commands and demonstrate their use in different network scenarios
2. Demonstrate the installation and configuration of network simulator.
3. Demonstrate and measure different network scenarios and their performance behavior.
4. Analyze the contents the packet contents of different protocols.
5. Implement the socket programming for client server architecture.
6. Design and setup a organization network using packet tracer.

Hardware Requirement:	**Software requirement:**
PC i3 processor and above	NS2.34, Protocol Analyzer (eg. Wireshark), Packet tracer (Eg. CISCO packet tracer)

Prerequisite: C Programming Language

Detailed syllabus:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Programming Language (C/java), Basic commands of windows and unix operating system, editor commands (eg nano/vi editor etc)	02	
I	Fundamentals of Computer Network	Understanding Basic networking Commands: Ping, Tracert, traceroute, ipconfig, ifconfig, nslookup, netstat	02	LO1
II	Basics of Network simulation	Installation and configuration of NS2 Introduction to Tcl Hello Programming	03	LO2
III	Simulation of Network Topology	Implementation of Specific Network topology with respect to 1. Number of nodes and physical layer configuration 2. Graphical simulation of network with Routing Protocols and traffic consideration (TCP, UDP) using NAM. 3. Analysis of network performance for quality of service parameters such as packet-delivery-ratio, delay and throughput 4. Comparative analysis of routing protocols with respect to QOS parameters using Xgraph/gnuplot for different load conditions.	05	LO3
IV	Protocol Analyzer	Installation of Wire shark Analysis of Packet headers,	04	LO4
V	Socket Programming	Socket Programming with C/Java 1.TCP Client, TCP Server	04	LO5

		2. UDP Client, UDP Server		
VI	Case study on designing network topology	A case study to design and configure any organization network eg. College network or campus network, using any packet tracer or network topology design software based on infrastructure requirements, servers and clients, traffic consideration and application requirements.	06	L06

Text Books:

1. Computer Network: Top Down approach, Behrouz Forouzan, Firoz Mossharraf. MGH
2. Packet analysis with Wire shark, Anish Nath, PACKT publishing

Reference Books:
1. NS2.34 Manual
2. Introduction to Network Simulator NS2, 2nd Edition, Teerawat Issariyakul, Ekram Hossain, Springer

Term Work:
Term Work shall consist of at least 10 to 12 practical's based on the above list. Also Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Oral Exam: An Oral exam will be held based on the above syllabus.

Course Code	Course Name	Theory	Practical	Tutorial	Theory	TW/Practical	Tutorial	Total
ITL402	Unix Lab	--	2	--	--	1	--	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Oral & Practical	Total
		Internal assessment			End Sem. Exam			
		Test 1	Test 2	Avg. of two Tests				

ITL402	Unix Lab	--	--	--	--	25	25	50

Lab Objectives: Students will try:

1. To introduce Basic Unix general purpose Commands
2. To learn network Unix commands.
3. To learn C programming in Unix editor environment.
4. To learn shell script and sed concepts.
5. To learn file management and permission advance commands.
6. To learn awk, grap, perl scripts.

Lab Outcomes: Student will be able to:

1. Identify the basic Unix general purpose commands.
2. Apply and change the ownership and file permissions using advance Unix commands.
3. Use the awk, grep, perl scripts.
4. Implement shell scripts and sed.
5. Apply basic of administrative task.
6. Apply networking Unix commands.

Prerequisite: C Programming Language and Operating System

Hardware requirement:

PC i3 and above.

Software requirement:

Unix, Editor, Bash shell, Bourne shell and C shell.

Detailed syllabus:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Programming syntax, Installation of Unix, concepts of operating system	02	--
I	Basic Commands	A brief history of UNIX, Unix Architecture. Logging into (and out of) UNIX systems, Changing your password, General format of UNIX commands. a) Installation of Unix operating system. b) User management in Unix. c) Study of Unix general purpose	04	LO1

		utility command list obtained from (cd, cp, ps, ls, mv, rm, mkdir, rmdir, man, who, cat, echo, more, date, time, kill, history, chmod, passwd, who am i, who, time, bc, history, clear, man, lost, chown, finger, pwd, cal, logout, shutdown) commands.		
II	Advance Commands	a) Study of Unix networking commands (ifconfig, ping, traceroute, netstat, nslookup, whois, hostname, tcpdump). b) Study of Unix file system (tree structure). c) Study of .bashrc, /etc/bashrc and Environment variables. d) Study File and directory permissions. e) Study of Editor Vi/other editor. f) Study of Bash shell, Bourne shell and C shell in Unix operating system.	04	LO1 LO2 LO5 LO6
III	Basic System administrative task	Process management Memory management File system management User management	04	LO1 LO2 LO5
IV	Shell scripts	a) Write a shell script program to display list of user currently logged in. b) Write a shell script program to display "HELLO WORLD". c) Write a shell script program to develop a scientific calculator. d) Write a shell Script program to check whether the given number is	04	LO1 LO4

		even or odd. e) Shell script Program to search whether element is present is in the list or not.		
V	Shell scripts and sed	a) Shell script program to check whether given file is a directory or not. b) Shell script program to count number of files in a Directory. c) Shell script program to copy contents of one file to another. d) Create directory, write contents on that and Copy to a suitable location in your home directory. e) Use a pipeline and command substitution to set the length of a line in file to a variable. f) Write a program using sed command to print duplicated lines of Input.	06	LO1 LO4
VI	grep, awk, perl scripts	a) Write a grep/egrep script to find the number of words character, words and lines in a file. b) Write an awk script to develop a Fibonacci series. c) Write a perl script to compute the power of a given number. d) Write an awk script to display the pattern of given string or number. e) Write a perl script to check a number is prime or not. f) Write an egrep script to display	04	LO1 LO2 LO3

		list of files in the directory.		

Text Books:

1. Unix, concepts and applications by Sumitabha Das, McGraw-Hill
2. Mastering Shell Scripting, Randal. K. Michael , Second Edition, Wiley Publication

References:

1. Unix Shell Programming by Yashwant Kanetkar
2. Unix shell programming by forozun

Term Work:
Term Work shall consist of at least 10 to 12 practical's based on the above list. Also Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Oral & Practical Exam: An Oral & Practical exam will be held based on the above syllabus.

Course Code	Course Name	Theory	Practical	Tutorial	Theory	TW/Practical	Tutorial	Total
ITL403	Microprocessor Programming Lab	--	2	--	--	1	--	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg. of two Tests				
ITL403	Microprocessor Programming Lab	--	--	--	--	25	25	50

Lab Objectives: Students will try to:

1. Learn assembling and disassembling of PC.
2. Get hands on experience with Assembly Language Programming.
3. Study interfacing of peripheral devices with 8086 microprocessor.
4. Understand techniques for faster execution of instructions and improve speed of operation and performance of microprocessors.
5. Learn fundamentals of designing embedded systems
6. Write and debug programs in TASM/MASM/hardware kits

Lab Outcomes: Students will be able to :

University of Mumbai, B. E. (Information Technology), Rev 2016   67

1. Apply the fundamentals of assembly level programming of microprocessors.
2. Build a program on a microprocessor using arithmetic & logical instruction set of 8086.
3. Develop the assembly level programming using 8086 loop instruction set.
4. Write programs based on string and procedure for 8086 microprocessor.
5. Analyze abstract problems and apply a combination of hardware and software to address the problem
6. Make use of standard test and measurement equipment to evaluate digital interfaces.

Prerequisite: Logic Design, Programming Languages(C, C++), COA

Hardware Requirement:

- Motherboard, RAM, Processor, Connectors, Cables, SMPS, HDD, Monitor, Graphics card (optional), Cabinet.
- 8086 microprocessor experiment kits with specified interfacing study boards.

Software Requirement:

- Microsoft Macro Assembler (TASM)/Turbo Assembler(TASM)

NOTE: Programs can be executed on assembler or hardware boards,

Detailed syllabus:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
I	PC Assembly	1. Study of PC Motherboard Technology (South Bridge and North Bridge). 2. Disassembling the System Unit & Identifying Internal Components and Connections. 3. Study of various connections and ports used in computer communication.	06	LO1
II	Arithmetic and logical operations in 8086 Assembly language programming	1. Program for 16 bit BCD addition 2. Program to evaluate given logical expression. 3. Convert two digit Packed BCD to Unpacked BCD. (any two)	04	LO2 LO6
III	Loop operations in 8086 Assembly language programming	1. Program to move set of numbers from one memory block to another. 2. Program to count number of 1's	06	LO3 LO6

		and 0;s in a given 8 bit number		
		3. Program to find the smallest/largest number from a given set of numbers.		
		4. Program to search for a given number		
		(any three)		
IV	String and procedure in 8086 Assembly language programming	1. Check whether a given string is a palindrome or not.	04	LO4 LO6
V	Procedure in 8086 Assembly language programming	1. Compute the factorial of a positive integer 'n' using recursive procedure. 2. Generate the first 'n' Fibonacci numbers. (any one)	02	LO4 LO6
VI	Interfacing with 8086 microprocessor	3. Interfacing Seven Segment Display 4. Interfacing keyboard matrix 5. Interfacing DAC (any two)	04	LO5 LO6

Text Books:

1. Scott Mueller,"Upgrading and repairing PCs", Pearson,

2. John Uffenbeck, "8086/8088 family: Design Programming and Interfacing:"Pearson Education

Reference Books:

1. K Bhurchandi, "Advanced Microprocessors & Peripherals", Tata McGraw-Hill Education

Term Work:
Term Work shall consist of at least 10 to 12 practical's based on the above list. Also Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Oral Exam: An Oral exam will be held based on the above syllabus.

Course Code	Course Name	Theory	Practical	Tutorial	Theory	TW/Practical	Tutorial	Total
ITL404	Python lab	--	2+2*	--	--	02	--	02

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Oral & Practical	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg. of two Tests				
ITL404	Python lab	--	--	--	--	50	50	100

*2 hours shown as practical's to be taken class wise lecture and other 2 hours to be taken as batch wise practicals in Lab.

Lab Objectives: The course will help the students to get familiar with:

1. Basics of Python programming
2. Decision Making and Functions in Python
3. Object Oriented Programming using Python
4. Files Handling in Python
5. GUI Programming and Databases operations in Python
6. Network Programming in Python

Lab Outcomes: Upon Completion of the course the learner should be able to:

1. Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python
2. Express different Decision Making statements and Functions
3. Interpret Object oriented programming in Python
4. Understand and summarize different File handling operations
5. Explain how to design GUI Applications in Python and evaluate different database operations
6. Design and develop Client Server network applications using Python

Hardware & Software Requirements:

Hardware Requirements	Software Requirements	Other Requirements
PC With following Configuration 1. Intel PIV Processor 2. 2 GB RAM 3. 500 GB Harddisk 4. Network interface card	1. Windows or Linux Desktop OS 2. Python 3.6 or higher 3. Notepad ++ 4.Python IDEs like Pydev, Netbeans or Eclipse 5. Mysql	1. Internet Connection for installing additional packages

Prerequisite Subjects: Structured Programming Approach & Java Programming

Detailed Syllabus:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Basic Programming syntax of Java/C. Installation and configuration of python.	02	
I	Basics of Python	**Theory**: Numbers in Python, Basic & Built-in Math functions, Number Formats, Strings, Quotes, print() Function, Assigning Values to Names & Changing Data Through Names, Copying Data, Tuples — Unchanging Sequences of Data, Lists — Changeable Sequences of Data, Dictionaries — Groupings of Data Indexed by Name, Special String Substitution Using Dictionaries , Arrays, Treating a String Like a List, Special Types, Ranges of Sequences, Working with Sets, Arrays. **Lab Experiment:** Write python programs to understand Expressions, Variables, Quotes, Basic Math operations, Strings: Basic String Operations & String Methods, List, Tuples, Dictionaries, Arrays. (Minimum Three Programs based on math operations, Strings and List/Tuples/ Dictionaries)	10	LO 1
II	Decision Making and Functions	**Theory:** If statement, if-elif-else, Repetition using while loop, for loop, break statement, Handling Errors- try: statement, except: statement, Functions-Grouping Code under a Name, defining a Function, describing a	10	LO 2

		function in the function, Checking & Setting Your Parameters, Calling Functions from within Other Functions, Functions Inside of Functions, Layers of Functions **Lab Experiment:** Write python programs to understand different decision making statements and Functions. (Minimum Three Programs based on Decision making, Looping Statements and Functions)		
III	Object Oriented Programming using Python programming	**Theory:** Creating a Class, Self Variables, Constructors, Types of Methods, Inner Classes, Constructors in Inheritance, Polymorphism,, The super() Method, Method Resolution Order (MRO), Operator Overloading, Method Overloading & Overriding, Interfaces in Python. Exceptions Handling: Errors in a Python Program, Exceptions, Exception Handling, Types of Exceptions, The Except Block, The assert Statement. Modules and Packages: Creating Modules and Packages, Documenting & Viewing Module, Basics of Testing Your Modules and Packages, Importing & exporting Modules. **Lab Experiment:** Write python programs to understand different Object oriented features in Python (Minimum four programs based on a) Classes & objects,	10	LO 3

		b) Constructors,		
		c) Inheritance & Polymorphism,		
		d) Exception handling		
IV	Files Handling	**Theory:** Types of Files in Python, Opening a File, Closing a File. Writing Text Files, Knowing Whether a File Exists or Not, Working with Binary Files, Appending Text to a File, Reading Text Files, File Exceptions, The with Statement Pickle in Python, Lambda and Filter, Map & range functions. **Lab Experiment:** Write python programs to understand different File handling operations	07	LO 4
V	GUI Programming and Databases	**Theory:** GUI Programming - Writing a GUI with Python: GUI Programming Toolkits, Creating GUI Widgets with Tkinter, Creating Layouts, Radio Buttons and Checkboxes, Dialog Boxes. Database Access - Python's Database Connectivity, Types of Databases Used with Python, Mysql database Connectivity with Python, Performing Insert, Deleting & Update operations on database **Lab Experiment:** Write python programs to understand GUI designing and database operations (Minimum Three programs based on GUI designing using Tkinter, Mysql database creation & Database connectivity with DML	07	LO 5

VI	Web Programming	**Theory:** Understanding Protocols, Introduction to Sockets, TCP/IP Server, TCP/IP Client, UDP Server, UDP Client, File Server, File Client, Two-Way Communication between Server and Client, Multithreaded Client-Server Chat Application **Lab Experiment:** Write python programs to understand TCP and UDP Sockets in Python (Minimum One programs based on TCP or UDP Sockets)	06	LO 6

Text Books:

1. James Payne, "Beginning Python: Using Python 2.6 and Python 3.1",Wrox Publication
2. Dr. R. Nageswara Rao,"Core Python Programming" , Dreamtech Press, Wiley Publication.
3. Magnus Lie Hetland,"Beginning Python From Novice to Professional", Second Edition", Apress Publication.

Reference Books:

1. Wesley J Chun," Core Python Applications Programming",Third Edition, Pearson Publication.
2. E. Balguruswamy," Introduction to Computing and Problem Solving using Python", McGraw Hill Publication
3. Learn to Master Python, from Star EDU solutions , by ScriptDemics

Term Work:
Term Work shall consist of at least 12 to 15 practical's based on the above list. Also Term work Journal must include at least 2 assignments.
Term Work Marks: 50 Marks (Total marks) = 40 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Oral & Practical Exam: An Oral & Practical exam will be held based on the above syllabus.