CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A)
Choice Based Credit System (with effect from 2018-19)
B.E (Information Technology)

SEMESTER – V

			Scheme of Instruction		Scheme of Examination			
S.No	Course Code	Title of the Course	Hours per week		Duration of SEE in Hours	Maximum Marks		Credits
			L/T	P/D		CIE	SEE	
THEORY								
1	16ITC16	Principles of Operating Systems	3	-	3	30	70	3
2	16ITC17	Database Systems	3/1	-	3	30	70	4
3	16ITC18	Software Engineering	3	-	3	30	70	3
4	16ITC19	Web Technology	3	-	3	30	70	3
5	16ITC20	Theory of Automata	3/1	-	3	30	70	4
		Elective - I	3	-	3	30	70	3
PRACTICALS								
6	16ITC21	Operating Systems and Web Technology Lab	-	3	3	25	50	2
7	16ITC22	Database Systems Lab	-	3	3	25	50	2
8	16ITC23	Mini Project-III	-	2	-	50	-	1
TOTAL			**20**	**8**	**-**	**280**	**520**	**25**

L: Lecture T: Tutorial D: Drawing P: Practical
CIE - Continuous Internal Evaluation SEE - Semester End Examination

Elective-I

S.No.	Subject Code	Subject Name
1.	16ITE01	Python Programming
2.	16ITE02	UNIX and Shell Programming
3.	16ITE03	Scripting Languages

16ITC16

PRINCIPLES OF OPERATING SYSTEMS

Instruction	3L Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

This course is introduced to

1. Learn various services provided by an operating system.
2. Learn, what a process is and how processes are synchronized and scheduled.
3. Learn different approaches of memory management.
4. Familiarizewith the structure and organization of the file system.
5. Familiarize with Protection and security aspects of operating systems.

Course Outcomes:

Upon successful completion of this course, student will be able to

1. Understand the services of an operating system, inter process communication and multithreaded programming.
2. Identify suitable process scheduling, deadlocks handling algorithms and solve process-synchronization problems.
3. Understand the organization of Main and Virtual memory in the operating system.
4. Understand File-System management.
5. Understand the Security problems, Threats and Protection mechanisms.
6. Choose an efficient algorithm based on different aspects for better performance of the system.

Prerequisites:

Computer Organization and Microprocessor (16ITC11), Programming and Problem Solving (16CSC01), Data Structures & Algorithms (16ITC02).

UNIT–I

Introduction: Definition of Operating System, Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security Kernel Data Structures Computing Environments , Open-Source Operating Systems .

Operating System Structures: Operating-System Services, User and Operating-System InterfaceSystem Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating-System Structure, System Boot.

Process: Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication.

Threads: Overview, Multicore Programming, Multithreading Models, Threading Issues.

UNIT–II

Process Synchronization: Background, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors.

CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Multiple-Processor Scheduling, Algorithm Evaluation.

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

UNIT–III

Memory: Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table.

Virtual Memory: Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files, Allocating Kernel Memory.

UNIT–IV

Mass-Storage Structure: Overview of Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap-Space Management

File-System Interface: File Concept, 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, Efficiency and Performance.

UNIT–V

Protection: Goals of Protection, Principles of Protection, Domain of Protection Access Matrix, Implementation of the Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems.

Security: The Security Problem, Program Threats, System and Network Threats, Cryptography as a Security Tool, User Authentication.

Text Book:

1. Abraham Silberschatz, Peter Galvin, Greg Gagne, "Operating System Concepts", Ninth Edition, John Wiley and sons publications, 2013.

Suggested Reading:

1. A.Tanenbaum, "Modern Operation Systems", Third Edition, Pearson Education, 2008.
2. William Stallings, "Operating Systems", Fifth Edition, Pearson Education, 2005.
3. Ida M.Flynn, "Understanding Operating Systems", Sixth Edition, Cengage, 2011.
4. D.M.Dhamdhere,"Operating systems a concept based approach", SecondEdition, McGraw-Hill, 2007.
5. Pramod Chandra P.Bhatt, "An Intoduction to Opearting Systems concepts and practice", Third Edition, PHI, 2014.

Web Resources:

1. http://nptel.ac.in/downloads/106108101/
2. http://www2.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/
3. http://www.cs.kent.edu/~farrell/osf03/oldnotes/

16ITC17

DATABASE SYSTEMS

Instruction	3L+1T Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	4

Course Objectives:

This course is introduced to

1. Familiarize with the fundamental concepts and the role of a database system in an organization.
2. Acquire knowledge on different issues in the design and implementation of a database system.
3. Learn how to write simple and moderately advanced database queries using SQL.
4. Learn logical database design and various database models.
5. Study the concepts of database security, concurrency and recoverability.

Course Outcomes:

Upon successful completion of this course, student will be able to

1. Understand the purpose of database systems and Design any domain specific database using E-R model by considering all the constraints and issues in the related domain.
2. Design and implement a database for any specified domain according to the well-known Relational data model and formulate Relational algebra expressions.
3. Use SQL for efficient data retrieval queries, advanced SQL concepts to access databases from programming languages and define various triggers to ensure the consistency of the databases.
4. Understand and apply normalization concepts in the design of a relational database.
5. Efficiently organize and manage data using indexing and hashing concepts to achieve good data retrieval performance.
6. Understand the concept of a database transaction and related database facilities, including concurrency control, backup and recovery, and data object locking and protocols**.**

Prerequisites:

Data Structuresand Algorithms (16ITC02), Java programming (16ITC10)

UNIT-I

Introduction: Database-System Applications, Purpose of Database Systems, View of Data, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and Information Retrieval Specialty Databases, Database Users and Administrators.

Database Design and the E-R Model: Overview of the Design Process, the Entity-Relationship Model, Constraints, Removing Redundant Attributes in Entity Sets, Entity-Relationship Diagrams, Reduction to Relational Schemas, Entity-Relationship Design Issues, Extended E-R Features, Alternative Notations for Modeling Data.

UNIT–II

Introduction to the Relational Model: Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Operations.

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Sub queries, Modification of the Database.

Intermediate SQL: Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Authorization.

UNIT–III

Advanced SQL: Accessing SQL from a Programming Language, Functions and Procedures, Triggers, Recursive Queries, Advanced Aggregation Features.

Relational Database Design: Features of Good Relational, Designs, Atomic Domains and First Normal Form, Decomposition using Functional Dependencies, Functional-Dependency Theory, Algorithms for Decomposition, Decomposition using Multivalued Dependencies.

UNIT-IV

Indexing and Hashing: Basic Concepts, Ordered Indices, B+ Tree Index Files, Multiple-Key Access, Static Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing, Bitmap Indices, Index Definition in SQL

Transactions: Transaction Concept, a Simple Transaction Model, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels.

UNIT–V

Concurrency Control: Lock-Based Protocols, Deadlock Handling, Multiple Granularity, Timestamp-Based Protocols, Validation-Based Protocols, Multiversion

Schemes, Snapshot Isolation, Insert Operations, Delete Operations and Predicate Reads.

Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with Loss of Nonvolatile Storage, ARIES, Remote Backup Systems.

Text Book:

1. Abraham Silberschatz, Henry F Korth, S. Sudarshan, “Database System Concepts”, Sixth Edition, McGraw-Hill International Edition, 2010.

Suggested Reading:

1. C J Date, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2003.
2. RamezElmasri, Shamkant B. Navathe, “Fundamentals of Database System”, Sixth Edition, Addison-Wesley, 2011.
3. Raghu Ramakrishnan,JohannesGehrke, “Database Management Systems”, Third Edition, McGraw-Hill International Edition, 2014.
4. Patric O’Neil, Elizabeth O’Neil, “Database-principles, programming and performance”, Second edition, Morgan Kaufmann Publishers, 2001.

Web Resources:

1. http://db-book.com/
2. https://www.tutorialspoint.com/dbms/
3. https://www.w3schools.in/dbms/
4. http://www.oracle-dba-online.com/sql/oracle_sql_tutorial.htm.
5. http://www.tutorialspoint.com/plsql/

16ITC18

SOFTWARE ENGINEERING

Instruction	3LHours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

This course is introduced to

1. Describe the various software life cycle models.
2. Explain the importance of the software development process.
3. Acquaint the students with software requirements and SRS document.
4. Familiarize the students with different software architectural styles.
5. Explain the importance of software quality and review techniques.

Course Outcomes:

Upon successful completion of this course, the students should be able to

1. Understand the nature of software and definition of software engineering, agile software development and agile process models.
2. Recognize the minimum requirements for the development of application.
3. Develop a system, component, or process to meet desired needs of a customer, conduct tests using various testing methods to verify and validate the results.
4. Involve in developing, maintain, efficient, reliable and cost effective software solutions.
5. Understand the risks, formulate and implement software projects.
6. Use the techniques, skills, and modern engineering tools necessary for engineering practice.

Prerequisites:

Programming and Problem Solving (16CSC01), Design and analysis of algorithms (16ITC08).

UNIT-I

Software and Software Engineering: The Nature of Software, Software Engineering. The Software Process, Software Engineering Practice.

A Generic view of Process : Software Engineering -A Layered Technology, A Process frame work, Process Models-Waterfall model, spiral model, The Unified Process, Product and Process, Process Assessment and Improvement, The CMMI, Agility: Introduction to Agile development, Product development in Internet time, Agile Process models-Scrum, Extreme programming, Agile Vs Waterfall Model. **Understanding Requirements:** Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Negotiating Requirements, Validating Requirements.

Requirements Modelling: Requirements Analysis, Scenario-Based Modeling, Problem Analysis, Data Flow Diagrams, Software Requirement and specifications, Behavioural and non-behavioural requirements.

UNIT-II

Design Concepts: Design within the Context of Software Engineering, The Design Process, Design Concepts. Cohesion & Coupling, Object Oriented Design-Identifying Objects and classes, User Interface Design.

Architectural Design: Software Architecture, Architecture Styles-pipe and filter architecture, black board architecture , layered architecture.

Component level Design: Designing Class Based Components, Conducting Component-Level Design, Designing Traditional Components, Component-Based Development.

UNIT-III

Quality Concepts: Software Quality, Achieving Software Quality.

Review Techniques: Cost Impact of Software Defects.

Software Quality Assurance: Background Issues, Elements of Software Quality Assurance, SQA Tasks, Goals and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, The SQA Plan.

UNIT-IV

Software Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Validation Testing, System Testing, The Art of Debugging. Testing Tools – Rational functional tester, Testing Standards, Selenium software testing tool.

Testing Conventional Applications: Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing, Control Structure Testing, Black–Box Testing, system testing, validation testing, beta testing, alpha testing, acceptance testing, regression testing,

Product Metrics: A Framework for Product Metrics, Size Metrics like LOC, Function points.

UNIT-V

Estimation: Observations on Estimation, the Project Planning Process, Software Scope and Feasibility, Resources, Software Project Estimation.

Risk Management: Reactive versus Proactive Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan.

Text Books:

1. Roger S.Pressman, "Software Engineering: A Practitioners Approach", 7th edition, McGrawHill, 2009.
2. Jim Highsmith, "Agile Software Development Ecosystems", Addison-Wesley 2002, ISBN 0201760436, 2010.

Suggested Reading:

1. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, "Fundamentals of Software Engineering", PHI, 2nd edition,2010.
2. Ali Behforoz and Frederic J.Hadson, "Software Engineering Fundamentals", Oxford End Press, 2010.
3. Pankaj Jalote, "An Integrated Approach to Software Engineering", 3rdedition, Narosa Publishing house, 2008.
4. James F.Peters, WitoldPedrycz, "Software Engineering-An engineering Approach", McGraw Hill, 2008.

Web Resources:

1. Software Engineering Sites: http://www.erg.abdn.ac.uk/users/brant/sre/ soft-eng.html.
2. SE web - Software Engineering Education Home Page: http:// tuvalu.cs.flinders.edu.au/seweb/se-ed/
3. ACM Classic Books Series: http://www.acm.org/classics/
4. Teaching Software Engineering - Lessons from MIT, by Hal Abelson and Philip Greenspun: http://philip.greenspun.com/teaching/teaching-software-engineering.
5. NASA Software Engineering Home Page:http://akao.larc.nasa.gov/dfc/ swreng.html
6. Software Engineering Hotlist at Georgia Tech: http:// www.cc.gatech.edu/computing/SW_Eng/hotlist.html
7. IEEE Guide to the Software Engineering Body of Knowledge: http:// www.swebok.org/

16ITC19

WEB TECHNOLOGY

Instruction	3L Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

This course is introduced to

1. Acquire knowledge about design and development of web pages.
2. Develop dynamic pages using Java Servlets and JSP.
3. Know about database connectivity and how it can be used in Web-based applications.
4. Describe the state of the art of frameworks.
5. ASP.NET, to tackle challenges that are simply out of reach on many other platforms.

Course Outcomes:

Upon successful completion of this course, student will be able to

1. Design responsive websites and validate web forms using JQuery.
2. Write a well-formed XML schemas and documents.
3. Develop dynamic web applications using Servlets and JSP.
4. Apply modern Framework techniques for web development to make applications maintainable.
5. Validate various types of controls.
6. Design and develop web applications using ASP.NET with Ajax based requests.

Prerequisites:

Java Programming (16ITC10)

UNIT-I

Introduction: Web Fundamentals, **HTML 5.0:** basic tags, Form controls, Layout Management, Graphics, Media, span and div tags.

Introduction to Cascading Style Sheets: CSS Selectors, CSS BOX Model, CSS Positioning, and CSS floating.

JQuery: Basics of JavaScript, JQuery syntax, Selectors, Events, JSON Fundamentals.

UNIT-II

Introduction to XML: The Syntax of XML, XML Document Structure, Document Type Definitions, Name Space, XML Schemas, Displaying XML Documents with CSS, XSLT Style Sheets and XML Processors.

Web Services: Web Service Architecture, structure and contents of SOAP message, structure of WSDL, Information in UDDI registry, UDDI Registry API.

UNIT-III

Java Servlets: Overview of Java Servlet API, Servlet Implementation, Servlet Configuration, Servlet Exceptions, Servlet Life cycle, Request and Response methods, Approaches to Session tracking, Request dispatching.

JSP Basics: Introduction to JSP, Directives, Scripting Elements, Standard Actions.

Database Connections: Introduction to JDBC, Database Drivers, JDBC API, connecting to my SQL, connecting to oracle, working with No SQL databases.

UNIT-IV

Struts framework: MVC Design pattern, Introduction to the modern web application framework, Architecture and flow of execution, working with actions and interceptors, Building the view with Tags and Results in detail, OGNL and type conversion, exploring the validation framework, writing the custom validator and Struts application development.

UNIT-V

ASP.NET: .Net framework, Web Form fundamentals-Html server controls, HTML control classes, Application events,ASP.net Configuration, Basic Web Control classes, State management, Building better web form - Validation, rich controls, user controls and graphics, ADO.NET Fundamentals, ASP.NET with Ajax.

Text Books:

1. Robert W.Sebesta, “Programming with World Wide Web”, Eighth Edition, Pearson Education, 2008.
2. John Pollak, “JQuery - A Beginners Guide”, McGraw-Hill Education, 2014.
3. Gustavo Alonso ,”Web Services: Concepts, Architectures and Applications” Springer Science & Business Media, 2004
4. Phil Hanna, “The Complete Reference JSP”, First Edition, Tata McGraw-Hill, 2003
5. Donald Brown, Chad Michael Davis, Scott Stanlick, “Struts 2 in Action”,Manning Publications, 2008.
6. Matthew MacDonald, “Beginning ASP.NET 4.5 in C#”, Illustrated, Apress, 2012.

Suggested Reading:

1. James Webber, SavasParastatidis, Ivan Robinson," Restin Practice: HyperMedid and System Architecture", First Edition,O'REILLY,2010.
2. Deitel, Deitel, Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2010.
3. SubramanyamAllamraju, "Professional Java Server programming", J2EE 1.3 Edition, CeditBuest, Apress Publications

Web Resources:

1. https://msdn.microsoft.com/en-us/library/office/ aa218647(v=office.11).aspx
2. https://sipb.mit.edu/iap/django/CCCDjango2010.pdf

16ITC20

THEORY OF AUTOMATA

Instruction	3L+1T Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	4

Course Objectives:

This course is introduced to

1. Study abstract computing models namely Finite Automata, Pushdown Automata, and Turning Machines.
2. Learn various grammars, formal languages and their relationships.
3. Learn the relation between various grammars and recognizers for different formal languages.
4. Evaluate and explain the differences between different computational models, such as Turing machines, push-down automata, finite automata, etc.
5. Familiarize with decidability and undecidability of computational problems.

Course Outcomes:

Upon successful completion of this course, student will be able to

1. Understand formal machines, languages and design Deterministic, Nondeterministic Finite automata for acceptance of languages.
2. Build regular expressions and their equivalent finite automata for different languages.
3. Define context-free grammars for certain languages and check the ambiguity of the grammars.
4. Design pushdown automata for accepting languages.
5. DesignTuring machines for computational problems, distinguish between decidability and undecidability.

Prerequisites:

Discrete Structures and Applications (16ITC01) and Data Structures and Algorithms (16ITC02).

UNIT-IAutomata: Introduction to Finite Automata, the Central Concepts of Automata Theory: Alphabets, Strings, and Languages.

Finite Automata: An Informal Picture Of Finite Automata: The Ground Rules, the Protocol, Enabling the Automata to Ignore Actions, the Entire System as an Automaton. Deterministic Finite Automata: Definition of a DFA, Simpler Notations for DFA's, Extending the Transition Function to Strings, The Language of a DFA, Nondeterministic Finite Automata: Definition of NFA, The Extended Transition Function, The Language of an NFA, Equivalence of NFA and DFA, An Application: Text Search, Finite Automata with Epsilon-Transitions: Use of €-transitions, The formal notation for an €- NFA, €-closure, Extended Transitions and Languages for €-NFA's, Eliminating €-transitions.

UNIT-II

Regular Expression and languages: Regular Expressions: The Operators of Regular Expressions, Building Regular Expressions. Finite Automata and Regular Expression: From DFAs to Regular Expressions, Converting DFA's to Regular Expressions by Eliminating States, Converting Regular Expressions to Automata, Applications of Regular Expressions, Algebraic Laws for Regular Expressions.

Properties of Regular Languages: Proving Languages not to be Regular: The pumping lemma for Regular Languages, Applications of Pumping Lemma, Closure Properties of Regular Languages, Decision Properties of Regular Languages: Testing Emptiness of Regular Languages, Testing Membership in a Regular Language. Equivalence and Minimization of Automata: Testing Equivalence of States, Testing Equivalence of Regular Languages, Minimization of DFA's.

UNIT-III

Context Free Grammars and Languages: Context-Free Grammars: Definition of Context Free Grammars, Derivations using a Grammar, Leftmost and Rightmost Derivation, The language of a Grammar, Parse Trees: Constructing Parse Trees, The Yield of a Parse Tree, Applications of CFGs, Ambiguity in Grammars and Languages: Ambiguous Grammars, Removing Ambiguity From Grammars, Leftmost Derivations as way to Express Ambiguity, Inherent Ambiguity.

Properties of Context Free Languages: Normal Forms for Context-Free Grammars: Eliminating Useless Symbols, Computing the Generating and Reachable Symbols, Eliminating Productions, Eliminating Unit Productions, Chomsky Normal Form, Greibachnormal form, Pumping Lemma for CFL's: Statement of the Pumping Lemma, Applications of Pumping Lemma for CFL's, Closure Properties of CFL's, Decision Properties of CFL's: Testing Emptiness of CFL's, Testing Membership in a CFL's.

UNIT-IV

Pushdown Automata: Definition of pushdown automaton: The Formal Definition of PDA, Graphical Notation for PDA's, Instantaneous Description of a PDA, The Language of a PDA: Acceptance by Final State, Acceptance by Empty Stack, From Empty Stack to Final State, From Final State to Empty Stack, Equivalence of PDA's and CFG's: From Grammars to PDA's, From PDA's to Grammars, Deterministic Pushdown Automata: Definition, Regular Languages and Deterministic PDA's, DPDA's to CFL's, DPDA's to Ambiguous Grammars.

UNIT-V

Introduction to Turing Machines: Problems that Computer Cannot Solve: The Turing Machine: Notation for the TM, Instantaneous Descriptions for TM's, Transitions Diagrams, The Language of a TM, Turing Machines and Halting, Programming Techniques for Turing Machines: Storage in the State, Multiple Tracks, Subroutines, Extensions to the Basic Turing Machine: Multitape Turing Machine, Equivalence of One-Tape and Multi-Tape TM's, Nondeterministic Turing Machines, Restricted Turing Machines: TM's with Sem infinite Tapes, Multistack Machines, Counter Machines. Turing Machine and Computers: Simulating a Computer by a TM.

Undecidability: A Language That Is Not Recursively Enumerable: Enumerating the Binary Strings, Codes for Turing Machines, The Diagonalization Language, An Undecidable problem that is RE: Recursive Languages, Compliments of Recursive and RE languages, The Universal Languages, Undecidablity of the Universal Language, Undecidable problems about Turing Machines: Reductions, TM's That Accept The Empty Language, Rice's Theorem and Properties of RE languages, Post's Correspondence Problem: Definition of PCP, The Modified PCP, Other Undecidable Problems.

Text Book:

1. John E. Hopcroft, Rajeev Motwani, Jeffery D Ullman, "Introduction to Automata Theory Languages and Computation", Third edition, Pearson Education, 2007.

Suggested Reading:

1. John C Martin. "Introduction to Language and Theory of Computation", 3rdedition, TMH, 2003.
2. Daniel Cohen, "Introduction to Computer Theory", 2ndedition, Wiley Publications, 2007.
3. Mishra K., Chandrasekaran N., "Theory of Computer Science (Automata, Languages and Computation)", 3rd edition, Prentice Hall of India 2008.
4. ShyamalendraKandar, "Introduction to Automata Theory, Formal Languages and Computation", Pearson, 2013.
5. Kamala Krithivasan, Rama R. "Introduction to Automata Theory, and Computation", Pearson 2009.

Web Resources:

1. http://nptel.ac.in/courses/106106049/
2. http://online.stanford.edu/course/automata-theory
3. https://www.tutorialspoint.com/automata_theory/

16ITE01

PYTHON PROGRAMMING
(Elective - I)

Instruction	3L Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

This course is introduced to

1. Familiarize the fundamentals of Python programming
2. Learn how to use lists, tuples, and dictionaries in Python programs
3. Learn how to read and write files in Python
4. Impart usage of exception handling in Python
5. Familiarize data visualization

Course Outcomes:

Upon successful completion of this course, student will be able to

1. Understand basic data structures of python
2. Understand the concepts of file I/O
3. Understand exception handling in Python.
4. Develop proficiency in creating GUI based applications
5. Plot data using appropriate Python visualization libraries
6. Develop simple Python applications.

Prerequisites:

Programming and Problem Solving (16CSC01), Programming Laboratory (16CSC02)

UNIT-I

Introduction to Python Programming: Using Python, The IDLE Programming Environment, Input and Output Processing, Displaying Output with the Print Function, Comments, Variables, Reading Input from the Keyboard, Performing Calculations, More About Data Output: New line, Item Separator, Escape Characters, Formatting parameters.

Decision Structures and Boolean Logic: if, if-else, if-elif-else Statements, Nested Decision Structures, Comparing Strings, Logical Operators, Boolean Variables. **Repetition Structures:** Introduction, while loop, for loop, Sentinels, Input Validation Loops, Nested Loops.

UNIT-II

Functions: Introduction, Defining and Calling a Function, Designing a Program to Use Functions, Local Variables, Passing Arguments to Functions, Global Variables and Global Constants, Value-Returning Functions Generating Random Numbers, Writing Our Own Value-Returning Functions, The math Module, Random Module, Time Module and Storing Functions in Modules.

UNIT-III

Lists and Tuples: Sequences, Introduction to Lists, List slicing, Finding Items in Lists with the in Operator, List Methods and Useful Built-in Functions, Copying Lists, Processing Lists, Two-Dimensional Lists, Tuples.

Strings: Basic String Operations, String Slicing, Testing, Searching, and Manipulating Strings.

Dictionaries and Sets: Dictionaries, Sets, Serializing Objects.

Recursion: Introduction, Problem Solving with Recursion, Examples of Recursive Algorithms.

UNIT-IV

Python File Input-Output: Opening and closing file, various types of file modes, reading and writing to files, manipulating directories

Exception Handling: What is exception, various keywords to handle exception such try, catch, except, else, finally, raise.

Regular Expressions: Concept of regular expression, various types of regular expressions, using match function.

UNIT-V

GUI Programming: Graphical User Interfaces, Using the tkinter Module, Display text with Label Widgets, Organizing Widgets with Frames, Button Widgets and Info Dialog Boxes, Getting Input with Entry Widget, Using Labels as Output Fields, Radio Buttons, Check Buttons.

Introduction to plotting in Python – Basic Plots- Line and Scatter Plot, Histograms and plotting data contained in files.

Text Books:

1. Tony Gaddis, "Starting Out With Python", 3rd edition, Pearson, 2015.
2. Charles Dierbach, "Introduction to Computer Science using Python", Wiley, 2013.

Suggested Reading:

1. Kenneth A. Lambert, "Fundamentals of Python", Delmar Cengage Learning, 2013.
2. James Payne, "Beginning Python using Python 2.6 and Python 3", wrox programmer to programmer, 2010.

3. Paul Gries, “Practical Programming: An Introduction to Computer Science using Python”,3rd edition, 2016.
4. Clinton W. Brownley, “Foundations for Analytics with Python”, 1st edition, O’Rielly Media, 2016.

Web Resources:

1. https://www.python.org/
2. https://www.coursera.org/learn/python
3. https://learnpythonthehardway.org/book/
4. https://www.coursera.org/specializations/python

16ITE02

UNIX AND SHELL PROGRAMMING
(Elective - I)

Instruction	3L Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

This course is introduced to

1. Familiarize students with the UNIX environment and basic UNIX utilities
2. Learn File systems and File structures.
3. Impart skills required to write shell scripts.
4. Develop skills required to formulate regular expressions.
5. Familiarize students with the routine system administrative features and tools.

Course Outcomes:

Upon successful completion of this course, the students should be able to

1. Understand the UNIX architecture, basics of vi editor and UNIX utilities.
2. Implement various File processing commands, change file permissions and directory permissions.
3. Create and manage processes using the knowledge of process attributes process creation and process control mechanisms.
4. Construct simple and complex shell scripts to automate jobs and processes in UNIX environment.
5. Locate and replace patterns at specific locations using regular expressions
6. Demonstrate administrator privileges, super user basic commands to add, modify and delete users.

Prerequisites: Programming and Problem Solving (16CSC01), Programming Laboratory (16CSC02).

UNIT-I

Introduction to Unix: The UNIX Operating System, The UNIX Architecture, Features of UNIX, Internal and External Commands, Command Structure, **General-Purpose Utilities:**cal, date, echo, printf, bc, script, mailx, passwd, who, uname, tty, sty,

The vi editor: vi Basics, Input Mode, Saving Text and Quitting, Navigation, Editing Text, Undoing Last Editing Instructions, Repeating the Last Command, Searching for a Pattern, Substitution.

UNIT-II

Handling Files: The File System, Parent Child Relationship, The HOME variable, pwd,cd, mkdir, rmdir, Absolute Pathnames, Relative Pathnames, The UNIX File System cat, cp, rm, mv, more, file, ls, wc, cmp, comm, diff,

Compressing and Archiving files: gzip and gunzip- Compressing and Decompressing files, tar- The Archival program, zip and unzip- Compressing and Archiving together.

File Attributes: ls options - l, -d,-lh,-la, File Ownership, File Permissions, chmod- Changing File permissions, Directory Permissions, Changing File ownership.

UNIT-III

The Shell: The Shells's interpretive Cycle, Shell Offerings, Pattern Matching, Escaping and quoting, Redirection, /dev/null and /dev/tty, Pipes, tee- Creating a tee, Command Substitution, Shell Variables.

The Process: Process Basics, ps- Process Status, System Processes (-e or –a), Mechanism of Process creation, Internal and External Commands, Process states and Zombies, Running jobs in Background, nice-Job Execution with low priority, Killing Processes with signals, Job Control, at and batch-Execute later, cron- Running jobs periodically, time-Timing Processes.

UNIT-IV

Simple Filters: pr- Paginating Files, head- Displaying the beginning of a File, tail-Displaying the end of a File, cut- Slitting a File vertically, paste-Pasting Files, sort-Ordering a File, uniq- Locate Repeated and Non-repeated Lines, tr-Translating Characters.

Filters using Regular Expressions: grep, Basic Regular Expressions, Extended Regular expressions, egrep, sed, Line Addressing, Using multiple instructions, Context Addressing, Writing Selected lines to a File, Text Editing, Substitution.

UNIT-V

Shell Programming: Shell scripts, read, Using Command Line Arguments, exit, The logical operators && and ||, Conditional execution- if, Using test and [ ] to evaluate expressions, case, expr, while, for, set and shift, trap, Debugging shell scripts with set–x.

System Administration: root, The administrator's privileges, Maintaining Security, User Management, Startup and Shutdown, Managing Disk Space, Device Files.

Text Book:

1. Sumitabha Das, "Unix Concepts and Applications", 4th Edition, TMH, 2006.

Suggested Reading:

1. Behrouz A. Forouzan, Richard F. Gilbery, "Unix and Shell Programming", 1st Edition, Cengage Learning India, 2003.

2. Graham Glass, King Ables, “Unix for programmers and users”, 3rd Edition, Pearson Education, 2009.
3. YashwanthKanitkar, “Unix Shell programming”, 1st Edition, BPB Publishers, 2010.

Web Resources:

1. http://openclassroom.stanford.edu/MainFolder/ CoursePage.php?course=PracticalUnix
2. https://www.shellscript.sh/
3. www.bash.academy/
4. http://linuxcommand.org/

16ITE03

SCRIPTING LANGUAGES
(Elective - I)

Instruction	3L Hours per week
Duration of End Examination	3 Hours
Semester End Examination	70 Marks
CIE	30 Marks
Credits	3

Course Objectives:

This course is introduced to

1. Write scripts to extract meaningful summaries from partially structured text.
2. Prepare students to use Python to perform common scripting tasks.
3. Allow students to use scikit-image library to learn image processing algorithms.
4. Familiarize students with PHP for making dynamic and interactive web pages.
5. Improve VB Scripting Skills for writing desktop, web applications and automation of tasks.

Course Outcomes:

Upon successful completion of this course, the students should be able to

1. Use Perl language features in web application development
2. Master the fundamentals of writing Python scripts
3. Implement algorithms and techniques involved in Digital Image Processing using scikit-image package
4. Gain the PHP programming skills needed to successfully build interactive, data-driven websites.
5. Use Ajax technology to load new content without leaving the current page, creating a better, faster experience for webpages
6. Develop web, desktop and various automation tasks using Visual Basic Scripting (VB Script)

Prerequisites:

Programming and Problem Solving (16ITC01), Data structures and algorithms (16ITC02)

UNIT-I

PERL- Names and Values, Variables, Scalars, Arrays and its operations, Hashes, Regular expressions, string manipulation, File management, Command line arguments, sub routines, Packages, Modules.

UNIT-II

Introduction to Python: Variables, Lists and Tuples, Introducing Functions , If statements, While Loops and Input, Basic Terminal Apps, Dictionaries, More Functions, Classes and OOPs, Exceptions.

UNIT-III

Simple Graphics and Image Processing using Python: “turtle” module; simple 2d drawing - colors, shapes; digital images, image file formats, image processing Simple image manipulations with ‘image’ module (convert to between, grey scale, blur, etc). Graphical user interfaces; event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames.

UNIT-IV

Programming with PHP: PHP Basics, String Manipulation and regular expressions, Form handling, Adding dynamic Content, Managing Web sessions, Handling Date & Time in PHP, Sending email with PHP, Object Oriented Programming and PHP7, Exception handling, Accessing Databases using PHP, AJAX with PHP.

UNIT-V

VBScript: Introduction to VBScript, Declaring and Using Variables, Operators, Operator Precedence and Constants, Using Conditional Statements, Loops in VBScript, Using Procedures and Functions, Arrays, Date Functions, Working with Strings and Cookies, Working with Events, Working with Excel Objects, Working with Connection Objects, Working with Files, Error Handling

Text Books:

1. Randal L. Schwartz, Tom Phoenix, brianfoy, “Learning Perl”, 5th Edition, O’Reilly Media, 2008.
2. Kenneth A. Lambert , “Fundamentals of Python First Programs”, Cengage Learning, 2012.
3. Luke Welling, Laura Thomson, “PHP and MySQL Web Development”, Pearson Education, 2017.
4. Kogent Solutions Inc, “Ajax Black Book”, Dreamtech press, 2008.
5. Adrian Kingsley-Hughes, Kathie Kingsley-Hughes, Daniel Read, “VBScript – Programmers Reference”, 3rd Edition, wiley publications, 2007.

Suggested Reading:

1. John ericsolem, “Programming Computer Vision with Python”, First edition, O’Reilly Media, 2012
2. Thomas A Powel, “The Complete Reference: AJAX”, 1st Edition, Tata McGraw Hill, 2008.

Web Resources:

1. https://docs.python.org/3/tutorial/
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/ 6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/
3. https://learn.perl.org/

16ITC21

OPERATING SYSTEMS AND WEB TECHNOLOGY LAB

Instruction	3L Hours per week
Duration of End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course Objectives:

This course is introduced to

1. Familiarize with various system calls of LINUX
2. Learn processes synchronization and scheduling algorithms
3. Design and develop web pages using html5, CSS positioning, servlets and JDBC.
4. Learn and write a well-formed XML schemas and documents.
5. Learn MVC based web application development using Struts2 and ASP.NET.

Course Outcomes:

Upon successful completion of this course, the students should be able to

1. Create multiple processes and replace a process image using different system calls.
2. Understand Inter-process communication using shared memory, message passing and pipes.
3. Analyze and evaluate different algorithms for CPU scheduling.
4. Design various web based applications using HTML5,Jquery and CSS.
5. Use JDBC, JSP and Struts 2 framework, to build modem web applications.
6. Design web site using ASP.NET with Ajax based requests.

Prerequisites:

Programming Laboratory (16CSC02), Data Structures and Algorithms Lab (16ITC05), Java Programming Lab (16ITC13).

List of Programs

1. Demonstrate the following system calls:
   a) fork b) execvp c) stat d) setenv&getenv
2. Implement Echo Server using
   a) Pipes b) Shared memory c)Message queues

3. Simulate the following CPU Scheduling Algorithm:
   a) FCFS b) SJF c) Round Robin
4. Implement Producer-Consumer Problem using
   a) Message passing b) Semaphores
5. Develop an e-commerce web site having the following specifications
   a) Use css for styling all the web controls.
   b) Use jquery for all form validations.
   c) All form submissions should be with AJAX only.
   d) Use menus in appropriate places.
6. Write a DTD and Schema for a library management system and give an XML example for each.
7. Build a java based dynamic working e-commerce website mentioned in question no.5 with database connections.
8. Develop a struts2 framework based “registration and login” application making use of validator framework.
9. Design and develop a simple web based application for “online quiz management” using ASP.NET.
10. Write an application to demonstrate data management using ADO.NET.

Text Books:

1. W. Richard Stevens, “Unix Network Programming”, Volume 2, 2nd edition, Pearson Education, 2015.
2. Robert W.Sebesta, “Programming with World Wide Web”, Eighth Edition, Pearson Education, 2008.
3. John Pollak, “JQuery - A Beginners Guide”, McGraw-Hill Education, 2014.
4. Phil Hanna, “The Complete Reference JSP”, First Edition, Tata McGraw-Hill, 2003.
5. Matthew MacDonald, “Beginning ASP.NET 4.5 in C#”, Illustrated, Apress, 2012.

Suggested Reading:

1. Silberschatz, Galvin, and Gagne,”Operating System Concepts”, 8thEdition,Wiley Publication.
2. Andrew S. Tanenbaum, “Modern Operating Systems”, 3rd Edition, GOAL Series.
3. James Webber, SavasParastatidis, Ivan Robinson,”Rest in Practice: HyperMedid and System Architecture”, First Edition,O’REILLY,2010.

4. Deitel, Deitel, Goldberg, “Internet & World Wide Web How To Program”, Third Edition, Pearson Education, 2010.
5. SubramanyamAllamraju, “Professional Java Server programming”, J2EE 1.3 Edition, CeditBuest, Apress Publications.

Web Resources:

1. http://www.tutorialspoint.com/unix/
2. https://msdn.microsoft.com/en-us/library/office/ aa218647(v=office.11).aspx

16ITC22

DATABASE SYSTEMS LAB

Instruction	3L Hours per week
Duration of End Examination	3 Hours
Semester End Examination	50 Marks
CIE	25 Marks
Credits	2

Course objectives:

This course is introduced to

1. Present the concepts and techniques relating to query processing.
2. Design and develop database for an application.
3. Learn the basic commands, SQL functions and the significance of triggers.
4. Learn how to manipulate a database using SQL.
5. Familiarize with the various methods of database security.

Course outcomes:

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

1. Design and implement database schemas by enforcing integrity constraints for a given problem domain.
2. Use SQL for database administration(to create tables, indexes, and views) and data manipulation.
3. Write efficient data retrieval queriesusing relational set operators and advanced SQL Join operators.
4. Do PL/SQL programming and define various triggers and cursors for the databases.
5. Create Security features and facilities for the database applications.
6. Design, create, and test data entry forms and detailed reports that require access to data in multiple tables.

Prerequisites:

Programming and Problem Solving (16CSC01)

List of Programs

1. Creation of database (Exercising commands like DDL and DML) (Note: use constraints while creating tables).
2. Exercising simple to complex queries.

a. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSECT Constraints.
b. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING clause and Creation and dropping of Views.
c. Exercising all types of Joins.

3. Demonstration of PL/SQL Blocks and Cursors.
4. Demonstration of Procedures and Functions.
5. Usage of Triggers (Programs using BEFORE and AFTER Triggers, Row and Statement level Triggers and INSTEAD OF Triggers).
5. Demonstrate Exception Handling by PL/SQL procedures for data validation.
6. Creating Password and Security features for applications.
7. Usage of File locking table locking, facilities in applications.
8. Creation of Forms and Generation of SQL reports.
9. Creation of full-fledged database application spreading over to 3 sessions.

Note:-The creation of sample database for the purpose of the experiments is to be pre-decided by the instructor.

Text Book:

1. Rick F Vander Lans, “Introduction to SQL”, Fourth edition, Pearson Education, 2007.

Suggested Reading:

1. Benjamin Rosenzweig, Elena Silvestrova, “Oracle PL/SQL by Example”, Fifth Edition, Pearson Education, 2015.
2. Albert Lulushi, “Oracle Forms Developer’s Handbook”, Pearson Education, 2006.

Web Resources:

1. http://www.oracle-dba-online.com/sql/oracle_sql_tutorial.htm.
2. https://www.javatpoint.com/sql-tutorial
3. https://www.tutorialspoint.com/sql/
4. http://www.tutorialspoint.com/plsql/
5. https://www.javatpoint.com/pl-sql-tutorial

16ITC23

MINI PROJECT - III

Instruction	2 Hours per week
Duration of End Examination	-
Semester End Examination	-
CIE	50 Marks
Credits	1

Course Objectives:

1. To enable students learn by doing.
2. To develop capability to analyse and solve real world problems.
3. To develop innovative ideas among the students.

Course Outcomes:

Students should be able to do the following:

1. To provide innovative solutions.
2. To work in a team.
3. To manage time and resources in the best possible manner.

The Students are required to implement one of the projects from project exercise given in the suggested readings of the theory subjects of the current semester / as suggested by the respective course faculty of that semester. During the implementation of the project, Personnel Software Process (PSP) has to be followed. Report of the project work has to be submitted for evaluation.