Scheme of Second Year

B.Tech Computer Science and Engineering (Cyber Security)
Credit Based Scheme of Studies/Examination
Semester III

S. N.	Course No.	Subject	L:T:P	Hours/Week	Credits	Examination Schedule (Marks)				Duration of Exam (Hrs)
						Major Test	Minor Test	Practical	Total	
1	PC-CS-CYS- 201A	Introduction to Cyber Security	3:0:0	3	3	75	25	0	100	3
2	PC-CS-CYS-203A	Data Structure	3:0:0	3	3	75	25	0	100	3
3	PC-CS-CYS-205A	Computational Thinking with Python	3:0:0	3	3	75	25	0	100	3
4	PC-CS-CYS-207A	Software Engineering	3:0:0	3	3	75	25	0	100	3
5	PC-CS-CYS-209A	Principles of Programming Languages	3:0:0	3	3	75	25	0	100	3
6	ES-CS-CYS- 211A	Computer Organization and Architecture	3:0:0	3	3	75	25	0	100	3
7	PC-CS-CYS- 213LA	Data Structure Lab	0:0:2	2	1	0	40	60	100	3
8	PC-CS-CYS- 215LA	Python Lab	0:0:2	2	1	0	40	60	100	3
9	PC-CS-CYS- 217LA	Software Engineering Lab	0:0:2	2	1	0	40	60	100	3
		Total		**24**	**21**	**450**	**270**	**180**	**900**	
10	SIM-201A*	Seminar on Summer Internship	2:0:0	2	0	0	50	0	50	

B.Tech Computer Science and Engineering (Cyber Security)
Scheme of Studies/Examination
Semester IV

S. N.	Course No.	Subject	L:T:P	Hours/ Week	Credits	Examination Schedule				Duration of Exam (Hrs.)
						Major Test	Minor Test	Practical	Total	
1	PC-CS-CYS-202A	Mathematics for Intelligent Systems	3:0:0	3	3	75	25	0	100	3
2	ES-CS-CYS-204A	Object-Oriented Programming System	3:0:0	3	3	75	25	0	100	3
3	PC-CS-CYS-206A	Data Base Management Systems	3:0:0	3	3	75	25	0	100	3
4	ES-CS-CYS-208A	Internet & Web technology	3:0:0	3	3	75	25	0	100	3
5	PC-CS-CYS- 210A	Operating System	3:0:0	3	3	75	25	0	100	3
6	PC-CS-CYS- 212A	Cryptographic Fundamentals	3:0:0	3	3	75	25	0	100	3
7	PC-CS-CYS- 214LA	Object Oriented Programming Lab	0:0:2	2	1	0	40	60	100	3
8	PC-CS-CYS- 216LA	Data Base Management Systems Lab	0:0:2	2	1	0	40	60	100	3
9	ES-CS-CYS- 218LA	Internet & Web Technology Lab	0:0:2	2	1	0	40	60	100	3
		Total		**24**	**21**	**450**	**270**	**180**	**900**	
10	MC-901A	Environmental Sciences	3:0:0	3		0		100	0	100

Syllabus Second Year

PC-CS-CYS-201A	**Introduction to Cyber Security**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	To gain a broad understanding in order to get predictive ways out related to cyber security.						
Course Outcomes							
CO1	To facilitate the basic knowledge of cyber security.						
CO2	To explore and sort issues related to different types of activities in cyber crime.						
CO3	To get enable to fix the various cyber attacks.						
CO4	To deal with the digital forensics and related scenarios of cyber crimes.						

Unit-I

Introduction: Introduction and Overview of Cyber Crime, Nature and Scope of Cyber Crime, Types of Cyber Crime: crime against individual, Crime against property, Cyber extortion, Drug trafficking, cyber terrorism. Need for Information security, Threats to Information Systems, Information Assurance, Cyber Security, and Security Risk Analysis.

Unit-II

Cyber Crime Issues: Unauthorized Access to Computers, Computer Intrusions, Viruses and Malicious Code, Internet Hacking and Cracking, Virus and worms, Software Piracy, Intellectual Property, Mail Bombs, Exploitation, Stalking and Obscenity in Internet, Password Cracking, Steganography, Key loggers and Spyware, Trojan and backdoors, phishing, DOS and DDOS attack, SQL injection, Buffer Overflow.

Unit-III

Introduction to cyber attacks: passive attacks, active attacks, Cyber crime prevention methods, Application security (Database, E-mail and Internet), Data Security Considerations-Backups, Archival Storage and Disposal of Data, Security Technology-Firewall and VPNs, Intrusion Detection, Access Control, Hardware protection mechanisms, OS Security

Unit-IV

Digital Forensics: Introduction to Digital Forensics, historical background of digital forensics, Forensic Software and Hardware, need for computer forensics science, special tools and techniques digital forensic life cycle, challenges in digital forensic.

Law Perspective: Introduction to the Legal Perspectives of Cybercrimes and Cyber security, Cybercrime and the Legal Landscape around the World, Why Do We Need Cyber laws, The Indian IT Act, Cybercrime Scenario in India, Digital Signatures and the Indian IT Act, Cybercrime and Punishment.

Suggested Books:

1. Nelson Phillips and Enfinger Steuart, “Computer Forensics and Investigations”, Cengage Learning, New Delhi, 2009.
2. Robert M Slade,” Software Forensics”, Tata McGraw - Hill, New Delhi, 2005.
3. Sunit Belapure and Nina Godbole, “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives”, Wiley India Pvt. Ltd.

PC-CS-CYS-203A	Data Structure						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	To introduce the principles and paradigms of Data Structures for design and implement the software systems logically and physically.						
Course Outcomes (CO)							
CO 1	To introduce the basic concepts of data structure , basic data types ,searching and sorting based on array data types.						
CO 2	To introduce the structured data types like Stacks and Queue and its basic operations' implementation.						
CO 3	To introduce dynamic implementation of linked list.						
CO 4	To introduce the concepts of Tree and graph and implementation of traversal algorithms.						

Unit-1

Introduction to Data Structures, Data Types, Built in and User Defined Data Structures, Applications of Data Structure, Algorithm Analysis, Worst, Best and Average Case Analysis, Notations of Space and Time Complexity, Basics of Recursion.

Arrays, One Dimensional Arrays, Two Dimensional Arrays and Multi-Dimensional Arrays, Sparse Matrices, Searching from array using Linear and Binary Searching Algorithm, Sorting of array using Selection, Insertion, Bubble, Radix Algorithm

Unit-2

Stacks: Definition, Implementation of Stacks and Its Operations, Evaluation of Infix, prefix and Postfix Expression, Inter-conversion of Infix, Prefix and Post-Fix Expression, Implementation of Merge Sort and Quick Sort Algorithm.

Queues: Definition, Sequential Implementation of Linear Queues and Its Operations, Circular Queue and Its Implementation, Priority Queues and Its Implementation, Applications of queues.

Unit-3

Linked Lists: Need of Dynamic Data Structures, Single Link List and Its Dynamic Implementation, Traversing, Insertion, Deletion Operations on Single Link Lists. Comparison between Static and Dynamic, Implementation of Linked List. Circular Link Lists and Doubly Link List, Dynamic Implementation of Primitive Operations on Doubly Linked Lists and Circular Link List. Dynamic Implementation of Stacks and Queues.

Unit-4

Trees: Definition, Basic Terminology, Binary Tree, External and Internal Nodes, Static and Dynamic Implementation of a Binary Tree, Primitive Operations on Binary Trees, Binary Tree Traversals: Pre-Order, In-Order and Post-Order Traversals. Representation of Infix, Post-Fix and Prefix Expressions using Trees.

Introduction to Binary Search Trees: B+ trees, AVL Trees, Threaded Binary trees, Balanced Multi-way search trees, Implementation of Heap Sort Algorithm.

Graphs: Basic Terminology, Definition of Undirected and Directed Graphs, Memory Representation of Graphs, Minimum-Spanning Trees, Warshal Algorithm, Graph Traversals Algorithms: Breadth First and Depth First.

Suggested Books:

- Theory and Problems of Data Structures by Jr. Symour Lipschetz, Schaum's outline, TMH.
- Data Structures and Algorithms by PAI, TMH.
- Fundamentals of Data structures by Ellis Horowitz and Sartaj Sahni, Pub, 1983, AW.
- Data Structures and Algorithms by A.V. Aho, J.E. Hopcroft and T.D. Ullman, Original edition, Addison-Wesley, 1999, Low Priced Edition.
- Data Structures and Program Design in C by Robert Kruse, PHI,
- Shukla, Data Structures using C++, Wiley India
- Introduction to Computers Science -An Algorithms Approach, Jean Paul Tremblay, Richard B. Bunt, 2002, T.M.H.
- Data Structure and the Standard Template library – Willam J. Collins, 2003, T.M.H.

Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.

PC-CS-CYS-205A	**Computational Thinking with Python**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	The purpose of this course is to introduce Python, a remarkably powerful dynamic programming language to write code for different operating systems along with application domain.						
Course Outcomes (CO)							
CO 1	To Learn python statements, comments and indentation, tokens, input and output methods using various example programs.						
CO 2	To Learn the different methods involved in List, String, Tuples and Dictionary and Design solutions for complex programs using decision making and looping statements.						
CO 3	To learn about different functions in python and Develop the function programs with all the concepts like lambda, decorators and generators.						
CO 4	To learn about the exception handling functions, file concepts and CSV and JSON.						

Unit-1

Introduction to python: Advantages of python programming-Tokens-Variables-Input/output methods-Data types- Operators

Unit-2

DATA STRUCTURES: Strings-Lists-Tuples-Dictionaries-Sets.
CONTROL STATEMENTS: Flow Control-Selection control Structure-if-if-else-if-elseif-else-Nested if iterative control structures-while loop, for loop and range.

Unit-3

FUNCTIONS: Declaration-Types of Arguments-Fixed arguments, variable arguments, keyword arguments and keyword variable arguments-Recursions-Anonymous functions: lambda- Decorators and Generators.

Unit-4

EXCEPTION HANDLING: Exception Handling-Regular Expression-Calendars and clock files: File input/output operations-Dictionary operations-Reading and writing in structured files: CSV and JSON.

Suggested Books:

- Bill Lubanovic, "Introducing Python Modern Computing in Simple Packages", 1st Edition, O'Reilly Media, 2014.
- Programming With Python Book ‘Himalaya Publishing House Pvt Ltd
- “Dive Into Python” by Mark Pilgrim
- Mark Lutz, “Learning Python”, 6th Edition, O'Reilly Media, 2014.
- David Beazley, Brian K. Jones, "Python Cookbook", 3rd Edition, O'Reilly Media, 2015.
- Mark Lutz, “Python Pocket Reference”, 6th Edition, O'Reilly Media, 2015.

Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.

PC-CS-CYS-207A	**Software Engineering**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	To gain a broad understanding of the discipline of software engineering and its application to the development and management of software process.						
Course Outcomes(CO)							
CO1	To understand the basic concepts of Software Engineering.						
CO2	To understand the fundamental concept of requirements engineering and Analysis Modelling.						
CO3	To understand the different design techniques and their implementation.						
CO4	To learn about software testing and maintenance measures.						

Unit-I

Introduction: Introduction to Software Engineering, Software Characteristics, Software Crisis, The Evolving role of Software, Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, RAD, V Model.

Unit-II

Software Requirement Specification: Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Data Flow Diagrams, Decision Tables, SRS Document, IEEE Standard for SRS.

Software Quality: Software Quality, Concept of Software Quality Assurance (SQA), SEI-CMM Model. Introduction to Software Risk Management and Software Configuration Management

Unit-III

Software Design: Basic Concept of Software Design, Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion.

Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design.

Software Measurement and Metrics: Various Size Oriented Measures: Halstead’s Software Science, Function Point (FP) Based Measures, COCOMO, Cyclomatic Complexity Measures: Control Flow Graphs.

Unit-IV

Software Construction: Software construction fundamentals, minimizing complexity, Top-Down and Bottom –Up programming, structured programming, Compliance with Design and Coding Standards.

Testing: Testing Objectives, Unit Testing, Integration Testing, system testing, Acceptance Testing, Regression Testing, Structural Testing, Functional Testing, debugging.

Maintenance: key issues, Types of software Maintenance, Cost of Maintenance, Software Re-Engineering.

Suggested Books:

1. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.
3. Pankaj Jalote, Software Engineering, Wiley India.
4. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
5. Ian Sommerville, Software Engineering, Addison Wesley.

PC-CS-CYS -209A	Principles of Programming Languages						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	To introduce the principles and paradigms of programming languages for design and implement the software intensive systems.						
Course Outcomes (CO)							
CO 1	To introduce the basic concepts of programming language, the general problems and methods related to syntax and semantics.						
CO 2	To introduce the structured data objects, subprograms and programmer defined data types.						
CO 3	To outline the sequence control and data control.						
CO 4	To introduce the concepts of storage management using programming languages.						

Unit-I: Introduction, Syntax and Semantics

Introduction: A brief history, Characteristics of a good programming language, Programming language translators- compiler and interpreters, Elementary data types – data objects, variable and constants, data types. Specification and implementation of elementary data types, Declarations, type checking and type conversions, Assignment and initialization, Numeric data types, enumerations, Booleans and characters.

Syntax and Semantics: Introduction, general problem of describing syntax, Formal method of describing Syntax, attribute grammar dynamic semantic.

Unit-II: Structured data objects, Subprograms and Programmer Defined Data Types

Structured data objects: Structured data objects and data types, specification and implementation of structured data types, Declaration and type checking of data structure, vector and arrays, records Character strings, variable size data structures, Union, pointer and programmer defined data objects, sets, files.

Subprograms and Programmer Defined Data Types: Evolution of data type concept abstraction, encapsulation and information hiding, Subprograms, type definitions, abstract data types, over loaded subprograms, generic subprograms.

Unit–III: Sequence Control and Data Control

Sequence Control: Implicit and explicit sequence control, sequence control within expressions, sequence control within statement, Subprogram sequence control: simple call return, recursive subprograms, Exception and exception handlers, co routines, sequence control. Concurrency – subprogram level concurrency, synchronization through semaphores, monitors and message passing

Data Control: Names and referencing environment, static and dynamic scope, block structure, Local data and local referencing environment, Shared data: dynamic and static scope, Parameter and parameter transmission schemes.

Unit-IV: Storage Management and Programming Languages

Storage Management: Major run time elements requiring storage, programmer and system controlled storage management and phases, Static storage management, Stack based storage management, Heap storage management, variable and fixed size elements.

Programming Languages: Introduction to procedural, non-procedural, structured, logical, functional and object oriented programming language, Comparison of C and C++ programming languages.

Suggested Books:

- Terrence W. Pratt, Marvin V. Zelkowitz, Programming Languages Design and Implementation, Pearson.
- Allen Tucker and Robert Noonan, Programming Languages–Principles and Paradigms, Tata McGraw-Hill, 2009.
- Ellis Horowitz, Fundamentals of Programming Languages, Galgotia Publications, 2010.
- C. Ghezzi, Programming Languages Concepts, Wiley Publications, 2010.

Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.

ES-CS-CYS-211A	**Computer Organization and Architecture**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	Student will be able to understand the basic concepts of computer architecture and organization, and understand the key skills of constructing cost-effective computer systems.						
Course Outcomes (CO)							
CO1	Be familiar with the functional units of the processor such as the register file and arithmetic-logical unit, and with the basics of systems topics						
CO2	Be familiar with the design trade-offs in designing and constructing a computer processor.						
CO3	Be familiar with the CPU design including the RISC/CISC architectures.						
CO4	Be familiar with the basic knowledge of I/O devices and interfacing of I/O devices with computer.						

Unit- I

Data representation and Computer arithmetic: Introduction to Computer Systems, Organization and architecture, evolution and computer generations; Fixed point representation of numbers, digital arithmetic algorithms for Addition, Subtraction, Multiplication using Booth's algorithm and Division using restoring and non restoring algorithms. Floating point representation with IEEE standards and its arithmetic operations.

Unit-II

Basic Computer organization and Design: Instruction codes, stored program organization, computer registers and common bus system, computer instructions, timing and control, instruction cycle: Fetch and Decode, Register reference instructions; Memory reference instructions. Input, output and Interrupt: configuration, instructions, Program interrupt, Interrupt cycle, Micro programmed Control organization, address sequencing, micro instruction format and microprogram sequencer.

Unit-III

Central Processing Unit: General register organization, stack organization, instruction formats, addressing modes, Data transfer and manipulation, Program control. CISC and RISC: features and comparison. Pipeline and vector Processing , Parallel Processing, Pipelining, Instruction Pipeline, Basics of vector processing and Array Processors.

Unit-IV

Input-output organization: I/O interface. I/O Bus and interface modules, I/O versus Memory Bus. Asynchronous data transfer: Strobe control, Handshaking, Asynchronous serial transfer. Modes of Transfer: Programmed I/O, Interrupt driven I/O, Priority interrupt; Daisy chaining, Parallel Priority interrupt. Direct memory Access, DMA controller and transfer. Input output Processor , CPU-IOP communication, I/O channel.

Suggested Books:

- William Stallings, "Computer Organization and Architecture – Designing for Performance", Sixth Edition, Pearson Education, 2003.
- Morris Mano, M., "Computer System Architecture," 3/e, Pearson Education, 2005.
- John P. Hayes, "Computer Architecture and Organization," 3/e, TMH, 1998.
- David A. Patterson and John L. Hennessy, "Computer Organization and Design: The Hardware/Software interface", Third Edition, Elsevier, 2005.
- V.P. Heuring, H.F. Jordan, "Computer Systems Design and Architecture", Second Edition, Pearson Education, 2004.
- Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", Fifth Edition, Tata McGraw Hill, 2002.

PC-CS-CYS-213LA	**Data Structure Lab**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Minor Test**	**Practical**	**Total**	**Time**
0	**0**	**2**	**1**	**40**	**60**	**100**	**2 Hours**
Purpose	To introduce the principles and paradigms of Data Structures for design and implement the software systems logically and physically.						
Course Outcomes (CO)							
CO1	Implement linear and non linear data structures using linked list.						
CO2	Apply various data structures such as stack, queue and tree to solve the problems.						
CO3	Implement various searching and sorting techniques.						
CO4	Choose appropriate data structure while designing the applications and analyze the complexity of the algorithms.						

LIST OF PRACTICALS:

1. Write a program for Binary search methods.
2. Write a program for insertion sort, selection sort and bubble sort.
3. Write a program to implement Stack and its operation.
4. Write a program for quick sort.
5. Write a program for merge sort.
6. Write a program to implement Queue and its operation.
7. Write a program to implement Circular Queue and its operation.
8. Write a program to implement singly linked list for the following operations: Create, Display, searching, traversing and deletion.
9. Write a program to implement doubly linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.

10 Write a program to implement circular linked list for the following operations: Create, Display, inserting, counting, searching, traversing and deletion.

11. Write a program to implement insertion, deletion and traversing in B tree

NOTE: A student has to perform at least ten experiments. Seven experiments should be performed from the above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.

PC-CS-CYS-215LA	**Python Lab**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Minor Test**	**Practical**	**Total**	**Time**
0	**0**	**2**	**1**	**40**	**60**	**100**	**2 Hours**
Purpose	The course is designed to provide Basic knowledge of Python						
Course Outcomes (CO)							
CO 1	To study fundamentals of python programming and implement basic programs.						
CO 2	To implement the searching technique using python.						
CO 3	To implement sorting techniques using python.						
CO 4	To implement matrix multiplication using python.						

LIST OF PRACTICALS

1. WAP to compute the GCD of two numbers.
2. WAP to find the square root of a number
3. WAP to find the Exponentiation (power of a number)
4. WAP to find the maximum of a list of numbers
5. WAP for Linear search and Binary search
6. WAP for Selection sort, Insertion sort
7. WAP for Merge sort
8. WAP to find first n prime numbers
9. WAP to multiply matrices
10. WAP that take command line arguments (word count)
11. WAP to find the most frequent words in a text read from a file
12. WAP to Simulate elliptical orbits in Pygame
13. WAP to Simulate bouncing ball using Pygame

Note: A student has to perform at least 10 experiments. At least seven experiments should be performed from the above list. Three experiments may be designed & set by the concerned institution as per the scope of the syllabus.

PC-CS-CYS-217LA	**Software Engineering Lab**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Minor Test**	**Practical**	**Total**	**Time**
0	**0**	**2**	**1**	**40**	**60**	**100**	**2 Hours**
Purpose	To gain a broad understanding of the discipline of software engineering implementation.						
Course Outcomes							
CO1	To learn about the reasons for the software crisis.						
CO2	To understand the software testing techniques.						
CO3	To understand the software metrics.						
CO4	To understand the different design techniques and their implementation.						

LIST OF PRACTICALS

1. To identify the role of the software in today’s world across a few significant domains related to day to day life.
2. To identify the problem related to software crisis for a given scenario.
3. To classify the requirement into functional and non-functional requirements.
4. To implement at least four software metrics.
5. Preparation of requirement document for standard application problems in standard format.(e.g Library Management System, Railway Reservation system, Hospital management System, University Admission system)
6. To prepare Project Schedule for standard application problems in standard format.
7. To implement the functional testing techniques.
8. To implement the structural testing techniques

PC-CS-CYS-202A	**Mathematics for Intelligent Systems**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	To introduce the principles and paradigms of linear algebra, differential equation and probability theory for modeling electrical, mechanical and computational experiments..						
Course Outcomes (CO)							
CO1	To develop an understanding of the basic concepts and techniques of linear algebra as applied to signal processing.						
CO2	To provide connection between the concepts of linear algebra, differential equation and probability theory.						
CO3	To enable the students to understand the use of calculus and Linear algebra in modelling electrical and mechanical elements.						
CO4	To equip the students to understand the role of probability theory in providing data sets for computational experiments in data science.						

Unit 1

Basics of Linear Algebra - Linear Dependence and independence of vectors - Gaussian Elimination - Rank of set of vectors forming a matrix - Vector space and Basis set for a Vector space - Dot product and Orthogonality - Rotation matrices - Eigenvalues and Eigenvectors and its interpretation - Projection matrix and Regression – Singular Value Decomposition.

Unit 2

Convolution sum, Convolution Integral, Ordinary Linear differential equations, formulation, analytical and Numerical solutions, Impulse Response Computations, formulating state space models of Physical systems.

Unit 3

Examples of ODE modelling in falling objects, satellite and planetary motion, Electrical and mechanical systems. Multivariate calculus, Taylor series, Introduction to Optimization.

Unit 4

Introduction to Probability Distributions and Monte Carlo Simulations.

Suggested Books :-

1. Gilbert Strang, Linear Algebra and Learning from Data, Wellesley, Cambridge press, 2019.
2. William Flannery, Mathematical Modelling and Computational Calculus, Vol-1, Berkeley Science Books, 2013.

ES-CS-CYS-204A	**Object-Oriented Programming System**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System.						
Course Outcomes (CO)							
CO1	To introduce the basic concepts of object oriented programming language and the its representation.						
CO2	To allocate dynamic memory, access private members of class and the behavior of inheritance and its implementation.						
CO3	To introduce polymorphism, interface design and overloading of operator.						
CO4	To handle backup system using file, general purpose template and handling of raised exception during programming.						

Unit–1

Introduction to C++, C++ Standard Library, Illustrative Simple C++ Programs. Header Files, Namespaces, Application of object oriented programming.

Object Oriented Concepts, Introduction to Objects and Object Oriented Programming, Encapsulation, Polymorphism, Overloading, Inheritance, Abstract Classes, Accessifier (public/ protected/ private), Class Scope and Accessing Class Members, Controlling Access Function, Constant, Class Member, Structure and Class

Unit-2

Friend Function and Friend Classes, This Pointer, Dynamic Memory Allocation and Deallocation (New and Delete), Static Class Members, Constructors, parameter Constructors and Copy Constructors, Deconstructors,

Introduction of inheritance, Types of Inheritance, Overriding Base Class Members in a Derived Class, Public, Protected and Private Inheritance, Effect of Constructors and Deconstructors of Base Class in Derived Classes.

Unit-3

Polymorphism, Pointer to Derived class, Virtual Functions, Pure Virtual Function, Abstract Base Classes, Static and Dynamic Binding, Virtual Deconstructors.

Fundamentals of Operator Overloading, Rules for Operators Overloading, Implementation of Operator Overloading Like <<,>> Unary Operators, Binary Operators.

Unit-4

Text Streams and binary stream, Sequential and Random Access File, Stream Input/ Output Classes, Stream Manipulators.

Basics of C++ Exception Handling, Try, Throw, Catch, multiple catch, Re-throwing an Exception, Exception specifications.

Templates: Function Templates, Overloading Template Functions, Class Template, Class Templates and Non- Type Template arguments.

Suggested Books:

1. The complete reference C ++ by Herbert shieldt Tata McGraw Hill.
2. Object Oriented Programming in Turbo C++ by Robert Lafore, 1994, The WAITE Group Press.
3. Shukla, Object Oriented Programming in c++, Wiley India.
4. C++ How to Program by H M Deitel and P J Deitel, 1998, Prentice Hall.
5. Programming with C++ By D Ravichandran, 2003, T.M.H.

Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.

PC-CS-CYS-206A	**Data Base Management Systems**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	**To familiarize the students with Data Base Management system**						
	Course Outcomes						
CO1	To provide introduction to relational model.						
CO2	To learn about ER diagrams and SQL.						
CO3	To understand about the concept of functional dependencies.						
CO4	To understand about Query Processing and Transaction Processing.						

Unit-1

Introduction: Concept & Overview of DBMS, Advantages of DBMS over file processing system, Database Languages, Responsibilities of Database Administrator, Database Users, Three Schema architecture of DBMS & Data Independence, Data Models.

Entity-Relationship Model: Basic concepts, Mapping Constraints, Keys, Entity-Relationship Diagram, Weak Entity Sets, Extended E-R features: Specialization and Generalization.

Unit-2

The Relational Data Model & Algebra: Relational Model: Structure of relational Databases, Relational Algebra & various operations (Set operation, select, project, joins, division), Relational Calculus: Domain, Tuple.

Integrity Constraints & Introduction to SQL:-

Domain Constraints, Referential Integrity Constraints, Basic Structure & Concept of DDL, DML, DCL, Aggregate Functions, Null Values, Introduction to views, creating, modifying and deleting views.

Unit-3

Relational Database Design : Functional Dependency, Different anomalies in designing a Database., Normalization – 1NF, 2NF, 3NF, Boyce-CODD Normal Form, Normalization using multi-valued dependencies, 4NF, 5NF.

Unit-4

Transaction Processing Concept: Introduction to transaction processing, transaction model properties, serializability:-Serial, non-serial and Serializable Schedules, Conflict Serializability.

Concurrency Control: Need of concurrency control, Different concurrency control Techniques: locking based, timestamps based technique. Deadlock handling and Recovery Techniques:- Deferred update/ immediate update, shadow paging.

Suggested Books:

- Elmasri and Navathe , “Fundamentals of Database Systems” , Addision-Wesley,
- Silberschatz, and Korth ,”Database System Concepts”, McGraw-Hill

- Date , “An Introduction to Database Systems” ,Addison-Wesley,
- Bhattacharyya, “Database Management Systems” , Tata McGraw-Hill Publishing.
-

Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.

ES-CS-CYS-208A	**Internet & Web Technology**					
Lecture	**Tutorial**	**Practical**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**75**	**25**	**100**	**3 Hours**
Purpose	To gain a broad understanding of the discipline of Web engineering and its application to the development and management of Web Applications.					
Course Outcomes						
CO1	Learn the basic concepts of information and web architecture.					
CO2	Learn about the skills that will enable to design and build high level web enabled applications.					
CO3	Understand the applicability of Java Script as per current software industry standards.					
CO4	Acquaint the latest programming language for the implementation of object based and procedure based applications using Python.					

Unit-1

Information Architecture: The role of Information Architect, Collaboration and communication, Organizing information, organizational challenges, Organizing web sites and Intranets, Creating cohesive organization systems, designing navigation systems, types of navigation systems, Integrated navigation elements, designing elegant navigation systems, Searching systems, Searching your web site, designing the search interface, Indexing the right stuff, To search or not to search grouping content, conceptual design, High level Architecture Blueprint. Architectural Page Mockups, Design Sketches.

Unit-2

Introduction to XHTML and HTML5: Origins and Evolution of HTML and XHTML, Basic Syntax, Standard XHTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms, HTML5, Syntactic Differences between HTML and XHTML.

Cascading Style Sheets: Introduction, Levels of Style Sheets, Style Specification Formats, Selector Forms, Property Value Forms, Font Properties, List Properties, Color, Alignment of Text, Box Model, Background Images, Conflict Resolution.

Unit -3

Java Script: Overview of JavaScript, Object Orientation and JavaScript, General Syntactic Characteristics, Primitives, Operations, and Expressions, Screen Output and Keyboard Input, Control Statements, Object Creation and Modification, Arrays, Functions, Constructors, Pattern Matching Using Regular Expressions, Errors in Scripts

Unit -4

Python: Introduction to Python, Data Types and Expressions, Control Statements, Strings and Text Files, Lists and Dictionaries, Design with Functions, Design with Classes

Suggested Books:

- By Peter Morville, Louis Rosenfeld, “Information Architecture on the World Wide Web”, O'Reilly Media, 2006.
- Robert W. Sebesta, “Programming The World Wide Web”, Eight Edition, Pearson India, 2015.
- Kenneth A. Lambert, “The Fundamentals of Python: First Programs”, 2011, Cengage Learning.
- Thomas A Powell, “HTML The Complete Reference”, Tata McGraw Hill Publications.

PC-CS-CYS-210A	**Operating System**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	To familiarize the students with the basics of Operating Systems.						
Course Outcomes (CO)							
CO1	To understand the structure and functions of operating system.						
CO2	To learn about processes, threads and scheduling algorithms.						
CO3	To understand the principle of concurrency and the concept of deadlocks.						
CO4	To understand various memory management scheme and to study I/O management and file systems.						

Unit-1

Introduction: Introduction to OS. Operating system functions, Different types of O.S.: batch process, multi-programmed, time-sharing, real-time, distributed, parallel.

System Structure: Computer system operation, I/O structure, storage structure, storage hierarchy, different types of protections, operating system structure (simple, layered, virtual machine), O/S services, system calls.

Unit-2

CPU scheduling: scheduling criteria, preemptive and non-preemptive scheduling, scheduling algorithms, algorithm evaluation, multi-processor scheduling.

Threads: overview, benefits of threads, user, and kernel threads.

Process Management: Concept of processes, process states, process control, co-operating processes, inter-process communication.

Process Synchronization: background, critical section problem, critical region, synchronization hardware, Classical problems of synchronization, semaphores.

Unit-3

Deadlocks: Concept of deadlock, deadlock characterization, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.

Memory Management: background, logical vs. physical address space, contiguous memory allocation, paging, segmentation, segmentation with paging. Concept of fragmentation.

Virtual Memory: background, demand paging, concept of page replacement, page replacement algorithms, allocation of frames, thrashing.

Unit-4

File Systems: file concept, file organization and access methods, allocation methods, directory structure, free-space management I/O Management: I/O hardware, polling, interrupts, DMA, kernel I/O subsystem (scheduling, buffering, caching, spooling and device reservation) Disk Management: disk structure, disk scheduling (FCFS, SSTF, SCAN,C-SCAN) , disk reliability, diskPerformance parametersProtection and Security:Goals of protection and security, security attacks, authentication, program threats, system threats, threat monitoring.Case studies: UNIX file system, Windows file system

Suggested Books:

- Operating System Concepts”, Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, Wiley
- Operating systems: a concept based approach”, Dhananjay M. Dhamdhere, McGraw Hill .
- Operating Systems : Internals and Design Principles, William Stallings, Pearson
- Operating Systems Design and Implementation” ,(Prentice Hall Software Series) Andrew S Tanenbaum and Albert S Woodhull.
- Taub and Schilling, Principles of Communication Systems, TMH.
- Mithal G K, Radio Engineering, Khanna Pub.

Note: The Examiner will be given the question paper template and will have to set the question paper according to the template provided along with the syllabus.

PC-CS-CYS-212A	**Cryptographic Fundamentals**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**3.0**	**75**	**25**	**100**	**3 Hours**
Purpose	**To familiarize the students with fundamentals of cryptography.**						
	Course Outcomes						
CO1	To understand basics of Cryptography and Network Security.						
CO2	To be able to secure a message over insecure channel by various means..						
CO3	To learn about how to maintain the Confidentiality, Integrity and Availability of a data						
CO4	To understand various protocols for network security to protect against the threats in the networks.						

Unit-1

Introduction to Cryptography and Block Ciphers: Introduction to security attacks - services and mechanism - introduction to cryptography - Conventional Encryption: Conventional encryption model - classical encryption techniques - substitution ciphers and transposition ciphers – cryptanalysis – steganography - stream and block ciphers - Modern Block Ciphers: Block ciphers principals - Shannon's theory of confusion and diffusion - fiestal structure - data encryption standard(DES) - strength of DES - differential and linearcrypt analysis of DES - block cipher modes of operations - triple DES – AES.

Unit-2

Public key cryptography and Authentication requirements: Principles of public key crypto systems - RSA algorithm - security of RSA - key management – Diffle-Hellman key exchange algorithm - introductory idea of Elliptic curve cryptography.

Integrity checks and Authentication algorithms: MD5 message digest algorithm - Secure hash algorithm (SHA) Digital Signatures: Digital Signatures - authentication protocols - digital signature standards (DSS) - proof of digital signature algorithm - Authentication Applications: Kerberos and X.509 - directory authentication service - electronic mail security-pretty good privacy (PGP) - S/MIME.

Unit-3

Cryptography and Network Security: Introduction to the Concept of Security, Cryptographic Techniques, Computer-based Symmetric and Asymmetric Key Cryptographic Algorithms, Public Key Infrastructure (PKI), Internet Security Protocols, Network Security.

Database Security: Data management technologies, Information security, Information Management Technologies, Security policies, Policy enforcement & related issues, Design principles, Multi-level relational data models, Security impact on database function.

Unit-4

Intrusion detection: Defining Intrusion Detection, Security concepts intrusion Detection concept, determining strategies for Intrusion Detection, Responses, Vulnerability Analysis, Credentialed approaches, Technical issues.

Suggested Books:

- NETWORK SECURITY AND MANAGEMENT BRIJENDRA SINGH, 2007
- Firewalls Complete Firewalls Complete Marcus Goncalves, 1998
- Information Security Architecture: An Integrated Approach to Security in the Organization Information Security Architecture: An Integrated Approach to Security in the Organization Jan Killmeyer, 2000
- Fundamentals of Network Security Fundamentals of Network Security John E. Canavan, 2001
- A Practical Introduction to Enterprise Network and Security Management A Practical Introduction to Enterprise Network and Security Management Bongsik Shin, 2017

PC-CS-CYS-214LA	**Object Oriented Programming Lab**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Minor Test**	**Practical**	**Total**	**Time**
0	**0**	**2**	**1**	**40**	**60**	**100**	**2 Hours**
Purpose	To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System.						
Course Outcomes (CO)							
CO1	To familiarize with the class and objects						
CO2	To implement the concept of constructors						
CO3	To familiarize the concept of operator overloading						
CO4	To implement the concepts of Inheritance						

LIST OF PRACTICALS

1. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power ( ) that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.

2. A point on the two dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis and 5 units up the Y axis. The sum of two points can be defined as a new point whose X coordinate is the sum of the X coordinates of the points and whose Y coordinate is the sum of their Y coordinates. Write a program that uses a structure called point to model a point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point. Interaction with the program might look like this:

```
    Enter coordinates for P1: 3 4
    Enter coordinates for P2: 5 7
    Coordinates of P1 + P2 are : 8, 11
```

3. Create the equivalent of a four function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user wants to do another calculation. The response can be ‘Y’ or ‘N’. Some sample interaction with the program might look like this.

```
Enter first number, operator, and second number: 10/ 3
    Answer = 3.333333
    Do another (Y/ N)? Y
    Enter first number, operator, second number 12 + 100
    Answer = 112
    Do another (Y/ N) ? N
```

4. A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this:

- Enter your area code, exchange, and number: 415 555 1212
- My number is (212) 767-8900
- Your number is (415) 555-1212

5. Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results maybe a DM object or DB objects, depending on the units in which the results are required. The display should be in the format of feet and inches or metres and cenitmetres depending on the object on display.

6. Create a class rational which represents a numerical value by two double values- NUMERATOR and DENOMINATOR. Include the following public member Functions:

- constructor with no arguments (default).
- constructor with two arguments.
- void reduce( ) that reduces the rational number by eliminating the highest common factor between the numerator and denominator.
- Overload + operator to add two rational number.
- Overload >> operator to enable input through cin.
- Overload << operator to enable output through cout.

Write a main ( ) to test all the functions in the class.

7. Consider the following class definition

```
class father {
protected : int age;
public;
father (int x) {age = x;}
virtual void iam ( )
{ cout < < “I AM THE FATHER, my age is : ”<< age<< end1:}
```

Derive the two classes son and daughter from the above class and for each, define iam ( ) to write our similar but appropriate messages. You should also define suitable constructors for these classes. Now, write a main ( ) that creates objects of the three classes and then calls iam ( ) for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call iam ( ) through the pointer to demonstrate polymorphism in action.

8. Write a program that creates a binary file by reading the data for the students from the terminal. The data of each student consist of roll no., name (a string of 30 or lesser no. of characters) and marks.

9. A hospital wants to create a database regarding its indoor patients. The information to store include

a) Name of the patient
b) Date of admission
c) Disease
d) Date of discharge

Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).

10. Make a class **Employee** with a name and salary. Make a class **Manager** inherit from **Employee**. Add an instance variable, named department, of type string. Supply a method to **to String** that prints the manager's name, department and salary. Make a class **Executive** inherits from **Manager**. Supply a method **to String** that prints the string **"Executive"** followed by the information stored in the **Manager** superclass object. Supply a test program that tests these classes and methods.

11. Imagine a tollbooth with a class called toll Booth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called payingCar ( ) increments the car total and adds 0.50 to the cash total. Another function, called nopayCar ( ), increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the ESC kay should cause the program to print out the total cars and total cash and then exit.

12. Write a function called reversit ( ) that reverses a string (an array of char). Use a for loop that swaps the first and last characters, then the second and next to last characters and so on. The string should be passed to reversit ( ) as an argument. Write a program to exercise reversit ( ). The program should get a string from the user, call reversit ( ), and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase, "Able was I ere I saw Elba)".

13. Create some objects of the string class, and put them in a Deque-some at the head of the Deque and some at the tail. Display the contents of the Deque using the forEach ( ) function and a user written display function. Then search the Deque for a particular string, using the first That ( ) function and display any strings that match. Finally remove all the items from the Deque using the getLeft ( ) function and display each item. Notice the order in which the items are displayed: Using getLeft ( ), those inserted on the left (head) of the Deque are removed in "last in first out" order while those put on the right side are removed in "first in first out" order. The opposite would be true if getRight ( ) were used.

14. Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class account that stores customer name, account number and type of account. From this derive the classes

cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:

a) Accept deposit from a customer and update the balance.
b) Display the balance.
c) Compute and deposit interest.
d) Permit withdrawal and update the balance.
e) Check for the minimum balance, impose penalty, necessary and update the balance.
f) Do not use any constructors. Use member functions to initialize the class members.

15. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data( ) to initialize baseclass data members and another member function display_area( ) to compute and display the area of figures. Make display_area ( ) as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area.

Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles and used as follows:

Area of rectangle = x * y

Area of triangle = ½ * x * y

NOTE: A student has to perform at least ten experiments. Seven experiments should be performed from the above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.

PC-CS-CYS-216LA	**Database Management Systems Lab**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Minor Test**	**Practical**	**Total**	**Time**
0	**0**	**2**	**1**	**40**	**60**	**100**	**2 Hours**
Purpose	**To implement practically the various concepts of DBMS**						
	Course Outcomes						
CO1	To understand& Implement basic DDL commands.						
CO2	To learn & Implement DML and DCL commands.						
CO3	To understand the SQL queries using SQL operators.						
CO4	To understand the concept of relational algebra and implement using examples.						

LIST OF PRACTCALS

1. Create a database and write the programs to carry out the following operation:
- Add , Delete and modify a record in the database
- Generate queries
- Data operations
- List all the records of database in ascending order.

2. To perform various integrity constraints on relational database.

3. Create a database and perform the following operations:-
1. Arithmetic and Relational operations
2. Group by & having clauses
3. Like predicate for pattern matching in database

4. Create a view to display details of employees working on more than one project.

5. Create a view to display details of employees not working on any project.

6. Using two tables create a view which shall perform natural join, equi join, outer joins.

7. Write a procedure to give incentive to employees working on all projects. If no such employee found give app. Message.

8. Write a procedure for computing amount telephone bill on the basic of following conditions.
1. telephone rent Rs. 205 including first 105 free units.
2. if extra units>0 but <500 then rate is 80 paise per unit.
3. if extra units>500 then rate is Rs. 1.20 per unit.

For this purpose create a table with name, Phone No., No. of units consumed, bill amount of a customer.

9. Write a procedure for computing income tax of employee on the basic of following conditions:-
1. if gross pay<=40,000 then I.T rate is 0%.
2. if gross pay>40,000 but <60000 then I.T rate is 10%.
3. if gross pay>60,000 but <1,00,0000 then I.T rate is 20%.
4. if gross pay>1,00,0000 then I.T rate is 30%.

For this purpose create a table with name, ssn, gross salary and income tax of the employee.

10. Write trigger for before and after insertion, deletion and updation process.

NOTE: A student has to perform at least ten experiments.Seven experiments should be performed from the above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus

ES-CS-CYS-218LA	**Internet & Web Technology Lab**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Minor Test**	**Practical**	**Total**	**Time**
0	**0**	**2**	**1**	**40**	**60**	**100**	**2 Hours**
Purpose	To introduce the concepts of HTML5, JavaScript and Python.						
Course Outcomes (CO)							
CO1	Design webpages using HTML, JavaScript and CSS.						
CO2	Design and test simple function/program to implement Searching and sorting techniques using Python.						
CO3	Develop program in Java Script for pattern matching using regular expressions and errors in scripts.						
CO4	Design client-server based web applications.						

LIST OF PRACTCALS

1. Create your own page with your favorite hobbies using HTML, JavaScript and CSS.
2. Create a frameset in HTML that is divided into three sections. The frameset should have three zones.
    a. The Topmost section of the frameset should take up about just 15% of the browser window. Name this frame title.
    b. The middle section should be 75% of the browser window. Name this frame title.
    c. The lower section should be 10% of the browser window. Name this frame menu.
3. Create pages for each section. For the lowermost section, create page that loads the content into the middle section. The topmost section should contain a page describing the web page itself.
4. Create a web page, which displays the map of your country Link, each city /state on the image map, such that the respective HTML page of the city/state is displayed when the user selects an area.
5. Add the tickertape applet to your page by customizing it for the following settings:
    a. Increase the count by one.
    b. Accordingly update the message count.
    c. Change the text color to (237,192,171)
    d. Experiment with changing the scrolling speed.
    e. Customize the message text as per your page requirement.
6. Incorporate a quest book into the Diary Food Webpage and use Java Script to build validations into the form.
7. Use Cascading Style sheets (CSS) to modify the following:
    a. Change background.
    b. Change font type, face and color.
    c. Align Text.
    d. Remove underlines from hyperlinks.
8. Write the program for using JavaScript by using for – loops (through a block of code a number of times), for/in - loops (through the properties of an object), while - loops (through a block of code while a specified condition is true), do/while - loops (through a block of code while a specified condition is true).

9. Write a program in Java Script for the following:
    a. Copying, passing, and comparing by value
    b. Copying, passing, and comparing by reference
    c. References themselves are passed by value
10. Write program in Java Script for pattern matching using regular expressions and errors in scripts.
11. Write a Python function/program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is an equilateral triangle.
12. Write the Python functions for linear search, binary search, selection sort, Bubble Sort, Insertion Sort and converting Fibonacci to a linear algorithm.
13. Write program in Python using Lists and dictionaries, Control statements and Strings and text files.

MC-901A	**Environmental Sciences**						
Lecture	**Tutorial**	**Practical**	**Credit**	**Major Test**	**Minor Test**	**Total**	**Time**
3	**0**	**0**	**0**	**75**	**25**	**100**	**3 Hours**
Purpose	To learn the multidisciplinary nature, scope and importance of Environmental sciences.						
Course Outcomes (CO)							
CO1	The students will be able to learn the importance of natural resources.						
CO2	To learn the theoretical and practical aspects of eco system.						
CO3	Will be able to learn the basic concepts of conservation of biodiversity.						
CO4	The students will be able to understand the basic concept of sustainable development.						

Unit-1

The multidisciplinary nature of environmental studies, Definition, Scope and Importance, Need for public awareness, Natural Resources: Renewable and Non-Renewable Resources: Natural resources and associated problems.

(a) Forest Resources: Use and over-exploitation, deforestation, case studies. Timber extraction, mining, dams and their effects on forests and tribal people.

(b) Water Resources: Use and over-utilization of surface and ground water, floods, drought, conflicts over water, dams-benefits and problems.

(c) Mineral Resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies.

(d) Food Resources: World Food Problems, changes caused by agriculture and overgazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies.

(e) Energy Resources: Growing energy needs, renewable and non-renewable energy sources, use of alternate energy sources. Case studies.

(f) Land Resources: Land as a resource, land, degradation, man induced landslides, soil erosion and desertification.

Role of an individual in conservation of natural resources, Equitable use of resources for sustainable lifestyle.

Unit-2

Ecosystem-Concept of an ecosystem: Structure and function of an ecosystem, Producers, consumers and decomposers, Energy flow in the ecosystem, Ecological Succession, Food Chains, food webs and ecological pyramids. Introduction, types, characteristic features, structure and function of the following ecosystem: (a) Forest Ecosystem, (b) Grassland Ecosystem, (c) Desert Ecosystem and (d) Aquatic Ecosystems (ponds, streams, lakes, rivers, oceans, esturaries

Field Work: Visit to a local area to document Environment assets-river/forest/grassland/hill/mountain, Visit to a local polluted site-Urban /Rural Industrial/Agricultural, Study of common plants, insects and birds, Study of simple ecosystems-pond, river, hill, slopes etc. (Field work equal to 5 lecture hours).

Unit-3

Biodiversity and its conservation: Introduction, Definition: genetic, species and ecosystem diversity. Biogeographical classification of India. Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values. Biodiversity of global, National and local levels. India as a mega-diversity nation Hot spot of Biodiversity, Threats to biodiversity: Habitat loss, poaching of wild life, man-wildlife conflicts, Endangered and endemic species of India, Conservation of Biodiversity- In situ and Ex-Situ conservation of biodiversity.

Environmental Pollution Definition: Cause, effects and control measures of (a) Air Pollution (b) Water Pollution (c) Soil Pollution (d) Marine Pollution (e) Noise Pollution (f) Thermal Pollution (g) Nuclear Hazards

Solid waste management- cause, effects and control measures of urban and industrial wastes, Role of an individual in prevention of pollution, Pollution case studies, Disaster management: floods, earthquake, cyclone and landslides

Unit-4

Social Issues and the Environment: From unsustainable to sustainable development, Urban problems related to energy, Water conservation, rain water harvesting, watershed management. Resettlement and rehabilitation of people: Its problems and concerns, Case Studies: Environmental ethics-issues and possible solutions. Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case studies: Wasteland Reclamation, Consumerism and waste products, Environment Protection Act, Air (Prevention and Control of Pollution) Act, Water (Prevention and Control of Pollution) Act, Wildlife Protection Act, Forest Conservation Act, Issues involved in enforcement of environmental legislation, Public Awareness, Human population and the Environment, Population growth, variation among nations, Population explosion-Family Welfare Programme, Environment and human health. Human rights, Value Education, HIV/AIDS, Women and Child Welfare, Role of Information Technology in Environment and Human Health, Case Studies, Drugs and their effects; Useful and harmful drugs, Use and abuse of drugs, Stimulant and depression drugs, Concept of drug de-addiction, Legal position on drugs and laws related to drugs.

Suggested Books:

- Environmental Studies- Deswal and Deswal. Dhanpat Rai and Co.
- Environmental Science and Engineering Anandan, P. and Kumaravelan, R. 2009. Scitech Publications (India) Pvt. Ltd., India
- Environmental Studies. Daniels Ranjit R. J. and Krishnaswamy. 2013. Wiley India.
- Environmental Science- Botkin and Keller. 2012. Wiley , India

Note: The Examiner will be given the question paper template to set the question paper.