Department: Computer Science

Programme: B.Sc. Computer Science

Programme Outcomes:

1. Students build strong foundation of Computer science.
    a. Develops fundamental skills for solving computational problems.
    b. The Mathematics & Statistics inculcate research oriented acumen.
2. Students are introduced with emerging trends in gradual way (from Fy to TY)
3. Students get well groomed for the challenges of ICT industry.
4. Students will develop capabilities to design formulations of computing models and its applications in diverse areas.
5. Course like Android Developer Fundamentals as a skill enhancement catering to modern day needs of Mobile platforms and applications.
6. Students will develop understanding of concepts and techniques for data management along with concepts of database at advance level.
7. Students will learn maths subjects like Combinatorics and Graph Theory & Linear Algebra. Graph theory is rapidly moving into the mainstream mainly because of its applications in diverse fields which include can further open new opportunities in the areas of genomics, communications networks and coding theory, algorithms and computations and operations research.
8. Students will be introduced with concepts Physical Computing and IoT programming will definitely open future area as Embedded Engineer, involvement in IoT projects, Robotics and many more.
9. Students will gain knowledge about Data Management and Machine Intelligence specific subject such as Data Science, Information Retrieval, Cloud Computing and Game Programming etc.
10. Finally programme develop a student as part of theoretically strong, innovatively skilled and ethically responsible generation of computer science professionals.
11. Students will get career opportunity in various fields such as software developer, web developer, software tester, game programmer, data analyst, network administrator etc.

Class: FYCS **Semester: I**

Course: USCS101 **Computer Organization and Design (Credits : 2 Lectures/Week: 3)**

Unit	Syllabus	Outcomes
I	**Computer Abstractions and** **Technology**: Basic structure and operation of a computer, functional units and their interaction. Representation of numbers and characters. **Computer Number System**: Binary , Octal, Hexadecimal Conversions & arithmetic. **Logic circuits and functions:** Combinational circuits and functions: Basic logic gates and functions, truth tables; logic circuits and functions. Minimization with Karnaugh maps. Synthesis of logic functions with and-or-not gates, nand gates, nor gates. Fan-in and fan-out requirements; tristate buffers. Half adder, full adder, ripple carry adder. (Flip flops) Gated S-R and D latches, edge-triggered D latch. Shift registers and registers. Decoders, multiplexers. Sequential circuits and functions: State diagram and state table; finite state machines and their synthesis.	Students understood basic hardware and functional units of computers. Students learned the number systems of computers and their conversions. Students learned to design various digital circuits using logic gates.

II	**Instruction set architectures**: Memory organization, addressing and operations; word size, big-endian and little-endian arrangements. Instructions, sequencing. Instruction sets for RISC and CISC (examples Altera NIOS II and Freescale ColdFire). Operand addressing modes; pointers; indexing for arrays. Machine language, assembly language, assembler directives. Function calls, processor runtime stack, stack frame. Types of machine instructions: arithmetic, logic, shift, etc. Instruction sets, RISC and CISC examples.	Students understood types of processor instructions. Students learned all basic concepts like memory organization, stacks, addressing modes, subroutines, compilers, assemblers etc which will be helpful to them while learning programming languages.
III	**Basic Processor Unit:** Main components of a processor: registers and register files, ALU, control unit, instruction fetch unit, interfaces to instruction and data memories. Datapath. Instruction fetch and execute; executing arithmetic/logic, memory access and branch instructions; hardwired and micro programmed control for RISC and CISC. **Basic I/O:** Accessing I/O devices, data transfers between processor and I/O devices. Interrupts and exceptions: interrupt requests and processing.	Students learned in brief components of the processor. Students understood the data flow among the various components of the processor and how the processor executes various instructions. Students understood how the data is being transferred to and from input/output devices and the processor. They also understood what is interrupt and how the processor manages it.

Course: USCS102 Programming with Python- I (Credits : 2 ectures/Week: 3)

Unit	Syllabus	Outcomes
I	Reasons for Python as the learner's first programming language. Introduction to the IDLE interpreter (shell) and its documentation. Expression evaluation: similarities and differences compared to a calculator; expressions and operators of types int, float, boolean. Built-in function type. Operator precedence. Enumeration of simple and compound statements. The expression statement. The assert statement, whose operand is a boolean expression (values true or false). The assignment statement, dynamic binding of names to values, (type is associated with data and not with names); automatic and implicit declaration of variable names with the assignment statement; assigning the valueNone to a name. The del (delete) statement. Input/output with print and input functions. A statement list (semicolon-separated list of simple statements on a single line) as a single interpreter command. The import statement for already-defined functions and constants. The augmented assignment statement. The built-inhelp() function. Interactive and script modes of IDLE, running a script, restarting the shell. The compound statement def to define functions; the role of indentation for delimiting the body of a compound statement; calling a previously defined function. Compound data types str, tuple and list (enclosed in quotes, parentheses and brackets, respectively). Indexing individual elements within these types. Strings and tuples are immutable, lists are mutable. Built-in functions min, max, sum. Interactive solution of model problems, (e.g., finding the square root of a number or zero of a function), by repeatedly executing the body of a loop (where the body is a statement list).	1. Students understood the concepts of programming before actually starting to write programs. 2. Learned to apply problem solving techniques using syntactically simple Python language. 3. Students get well versed with python working environment (shell & IDLE). 4. Learned to find help about commands/ statements with in IDLE/shell. 5. Got knowledge about reusable codes – functions - Library & user defined
II	User defined function- def keyword. Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables.	1. Students got familiar with the basic constructs of programming such as data, operations, conditions, loops, functions etc. 2. Learned about compound data

	The range function, the iterative for statement. The conditional statements if, if-else, if-elif-else. The iterative statements while, while-else, for-else. The continue statement to skip over one iteration of a loop, the break statement to exit the loop. Nested compound statements. Dictionaries: concept of key-value pairs, techniques to create, update and delete dictionary items. Problem-solving using compound types and statements.	types & their use in applications.
III	Anonymous functions. List comprehensions with variety of examples. Gentle introduction to object-oriented programming; using the built-in dir() function, enumerate the methods of strings, tuples, lists, dictionaries. Using these methods for problem-solving with compound types.	1. Students got better understanding of advanced compound data types. 2. Student learned to define objects. 3. They tried to handle their own objects in program. 4. Got idea about object behavior.

Course: USCS103 Free and Open-source Software (Credits : 2 Lectures/Week: 3)

Unit	Syllabus	Outcomes
I	Introduction Introduction: Open Source, Free Software, Free Software vs. Open Source software, Public Domain Software, FOSS does not mean no cost. History: BSD, The Free Software Foundation and the GNU Project. Methodologies Open Source History, Initiatives, Principle and methodologies. Philosophy : Software Freedom, Open Source Development Model Licenses and Patents: What Is A License, Important FOSS Licenses (Apache,BSD,GPL, LGPL), copyrights and copy lefts, Patents Economics of FOSS : Zero Marginal Cost, Income-generation opportunities, Problems with traditional commercial software, Internationalization Social Impact Open source vs. closed source, Open source government, Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source, Open Source in Government.	1. Students got a good working knowledge of Open Source ecosystem, its use, impact and importance. 2. Various types of software's learned by students.
II	Case Studies Example Projects: Apache web server, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, wordpress, GCC, GDB, github, Open Office. Study: Understanding the developmental models, licensings, mode of funding,commercial/non-commercial use. Open Source Hardware, Open Source Design, Open source Teaching. Open source media. Collaboration, Community and Communication Contributing to Open Source Projects Introduction to github, interacting with the community on github, Communication and etiquette, testing open source code, reporting issues, contributing code. Introduction to wikipedia, contributing to Wikipedia Or contributing to any prominent open source project of student's choice. Starting and Maintaining own Open Source Project.	1.Students got a platform to present their work as part of the various sites. 2. Students learned about Open Source methodologies and ecosystem. 3. Students developed their own site.
III	Understanding Open Source Ecosystem Open Source Operating Systems: GNU/Linux, Android, Free BSD, Open Solaris. Open Source Hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, debuggers, Programming languages, LAMP, Open Source database technologies	1. Students gain knowledge of various technologies. 2. Students gain knowledge of open source databases.

Course: **USCS104 Database Systems (Credits : 2 Lectures/Week: 3)**

Unit	Syllabus	Outcomes
I	**Introduction to DBMS –** Database, DBMS – Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture **Data models** - Client/Server Architecture, Object Based Logical Model, Record Based Logical Model (relational, hierarchical, network) **Entity Relationship Model -** Entities, attributes, entity sets, relations, relationship sets, Additional constraints (key constraints, participation constraints, weak entities, aggregation / generalization, Conceptual Design using ER (entities VS attributes, Entity Vs relationship, binary Vs ternary, constraints beyond ER) **Relational data model**– Domains, attributes, Tuples and Relations, Relational Model Notation, Characteristics of Relations, Relational Constraints - primary key, referential integrity, unique constraint, Null constraint, Check constraint **ER to Table**- Entity to Table, Relationship to tables with and without key constraints.	1. Students understood how to evaluate business information problem and find the requirements of a problem in terms of data. 2. Student got familiar with the concepts of DBMS with respect to the relational model 3. Students understood how to specify the functional and data requirements for a typical database application and also understood creation, manipulation and querying of data in databases.
II	**Schema refinement and Normal forms:** Functional dependencies, first, second, third, and BCNF normal forms based on primary keys, lossless join decomposition. **Relational Algebra** operations (selection, projection, set operations union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division) **DDL Statements** - Creating Databases, Using Databases, datatypes, Creating Tables (with integrity constraints – primary key, default, check, not null), Altering Tables, Renaming Tables, Dropping Tables, Truncating Tables, Backing Up and Restoring databases **DML Statements** – Viewing the structure of a table insert, update, delete, Select all columns, specific columns, unique records, conditional select, in clause, between clause, limit, aggregate functions (count, min, max,	1. Students were able to design the database schema with the use of appropriate data types for storage of data in database. 2. Students learned the various ways of creating database using ddl, dml. 3. Internal working of query, optimization techniques how a query is evaluated using relational algebra.

	avg, sum), group by clause, having clause	
III	**Functions** – String Functions (concat, instr, left, right, mid, length, lcase/lower, ucase/upper, replace, strcmp, trim, ltrim, rtrim), Math Functions (abs, ceil, floor, mod, pow, sqrt, round, truncate) Date Functions (adddate, datediff, day, month, year, hour, min, sec, now, reverse) **Joining Tables** – inner join, outer join (left outer, right outer, full outer) **Subqueries** – subqueries with IN, EXISTS, subqueries restrictions, Nested subqueries, ANY/ALL clause, correlated subqueries **Database Protection:** Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control (Concept) **Views** (creating, altering dropping, renaming and manipulating views) **DCL Statements (theoretical)-** creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges.	1. Students will be able to create, manipulate, query and back up the databases. 2. Students learned different types of functions. 3. They also learned how to work with two tables, join a table, who is a dba.

Course: **USCS105 Discrete Mathematics (Credits : Lectures/Week: 3)**

Unit	Syllabus	Outcomes
I	**Recurrence Relations** **(a) Functions:** Definition of function. Domain, co domain and the range of a function. Direct and inverse images. Injective, surjective and bijective functions. Composite and inverse functions. **(b) Relations**: Definition and examples. Properties of relations , Partial Ordering sets, Linear Ordering Hasse Daigrams , Maximum and Minimum elements, Lattices **Recurrence Relations:** Definition of recurrence relations, Formulating recurrence relations, solving recurrence relations- Back tracking method, Linear homogeneous recurrence relations with constant coefficients. Solving linear homogeneous recurrence relations with constant coefficients of degree two when characteristic equation has distinct roots and only one root, Particular	1. Students were familiarize with mathematical structures that are discrete. 2. Students learned about - sets and functions -forming and solving recurrence relations and different counting principles. 3. These concepts are useful to study or describe objects or problems in computer algorithms and programming languages.

	solutions of non linear homogeneous recurrence relation, Solution of recurrence relation by the method of generation functions, Applications- Formulate and solve recurrence relation for Fibonacci numbers, Tower of Hanoi, Intersection of lines in a plane, Sorting Algorithms.	
II	**Counting Principles , Languages and Finite State Machine** **(a) Permutations and Combinations:** Partition and Distribution of objects, Permutation with distinct and indistinct objects, Binomial numbers, Combination with identities: Pascal Identity, Vandermonde's Identity, Pascal triangle, Binomial theorem, Combination with indistinct objects. **(b) Counting Principles:** Sum and Product Rules, Two-way counting, Tree diagram for solving counting problems, Pigeonhole Principle (without proof); Simple examples, Inclusion Exclusion Principle (Sieve formula) (Without proof). **Languages, Grammars and Machines:** Languages , regular Expression and Regular languages, Finite state Automata, grammars, Finite state machines, Gödel numbers, Turing machines.	Provide basic knowledge about models of automata theory and the corresponding formal languages.
III	**Graphs and Trees** **(a) Graphs :** Definition and elementary results, Adjacency matrix, path matrix, Representing relations using diagraphs, Warshall's algorithm- shortest path , Linked representation of a graph, Operations on graph with algorithms - searching in a graph; Insertion in a graph, Deleting from a graph, Traversing a graph- Breadth-First search and Depth-First search. **(b) Trees:** Definition and elementary results. Ordered rooted tree, Binary trees, Complete and extended binary trees, representing binary trees in memory, traversing binary trees, binary search tree, Algorithms for searching and inserting in binary search trees, Algorithms for deleting in a binary search tree.	Students understood concept of graphs and trees, which are widely used in software.

Course: USCS106 Subject: Descriptive Statistics and Introduction to Probability

(Credits : 2 Lectures/Week: 3)

Unit	Syllabus	Outcomes
I	Data Presentation Data types : attribute, variable, discrete and continuous variable Data presentation : frequency distribution, histogram o give, curves, stem and leaf display Data Aggregation Measures of Central tendency: Mean, Median, mode for raw data, discrete, grouped frequency distribution. Measures dispersion: Variance, standard deviation, coefficient of variation for raw data, discrete and grouped frequency distribution, quartiles, quantiles Real life examples (15 L)	Enable Students to know descriptive statistical concepts.
II	Moments: raw moments, central moments, relation between raw and central moments Measures of Skewness and Kurtosis: based on moments, quartiles, relation between mean, median, mode for symmetric, asymmetric frequency curve. Correlation and Regression: bivariate data, scatter plot, correlation, nonsense correlation, Karl pearson's coefficients of correlation, independence. Linear regression: fitting of linear regression using least square regression, coefficient of determination, properties of regression coefficients (only statement)	Students will be able to understand the relation and future prediction between two entities which are related with each other
III	Probability : Random experiment, sample space, events types and operations of events Probability definition : classical, axiomatic, Elementary Theorems of probability (without proof) – $0 \leq P(A) \leq 1$, – $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ – $P(A') = 1 - P(A)$ – $P(A) \leq P(B)$ if $A \subset B$ Conditional probability, 'Bayes' theorem, independence, Examples on Probability	Students will be able to Develop problem-solving techniques needed to accurately calculate **probabilities**. Apply problem-solving techniques to solving real-world events.

Course: USCS107 Soft Skills Development (Credits : 2 Lectures/Week: 3)

Unit	Syllabus	Outcomes
I	**Introduction to Soft Skills and Hard Skills** **Personality Development:** Knowing Yourself, Positive Thinking, Johari's Window, Communication Skills, Non-verbal Communication, Physical Fitness **Emotional Intelligence:** Meaning and Definition, Need for Emotional Intelligence, Intelligence Quotient versus Emotional Intelligence Quotient, Components of Emotional Intelligence, Competencies of Emotional Intelligence, Skills to Develop Emotional Intelligence **Etiquette and Mannerism:** Introduction, Professional Etiquette, Technology Etiquette **Communication Today:** Significance of Communication, GSC's 3M Model of Communication, Vitality of the Communication Process, Virtues of Listening, Fundamentals of Good Listening, Nature of Non-Verbal Communication, Need for Intercultural Communication, Communicating Digital World	1) To know about various aspects of soft skills and learn ways to develop personality 2) The students learned the concept of positive thinking and the tool of assessing their personality. 3) The students understood the EI concept and as well its importance. 4) Understood the importance and type of communication, behaviour in personal and professional environment. 5) The student is able to understand different modes of communication in today's digital world.
II	**Academic Skills** **Employment Communication:** Introduction, Resume, Curriculum Vitae, Scannable Resume, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter **Professional Presentation:** Nature of Oral Presentation, Planning a Presentation, Preparing the Presentation, Delivering the Presentation **Job Interviews:** Introduction, Importance of Resume, Definition of Interview, Background Information, Types of Interviews, Preparatory Steps for Job Interviews, Interview Skill Tips, Changes in the Interview Process, FAQ During Interviews **Group Discussion:** Introduction, Ambience/Seating Arrangement for Group Discussion, Importance of Group	1) Has provided insight into much needed technical and non-technical qualities in career planning. 2) Understood various issues in personal and professional communication and learned to overcome them. 3) The student is able to know resume, understand its various types and difference between CV & Resume. 4) The students learned how to prepare presentations and deliver them professionally. 5) Also they got tips for

	Discussions, Difference between Group Discussion, Panel Discussion and Debate, Traits, Types of Group Discussions, topic based and Case based Group Discussion, Individual Traits	making interview effective and also has help them to participate in GD.
III	**Professional Skills** **Creativity at Workplace:** Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at Work, The Six Thinking Hat Method **Ethical Values:** Ethics and Society, Theories of Ethics, Correlation between Values and Behavior, Nurturing Ethics, Importance of Work Ethics, Problems in the Absence of Work Ethics **Capacity Building: Learn, Unlearn and Relearn**: Capacity Building, Elements of Capacity Building, Zones of Learning, Ideas for Learning, Strategies for Capacity Building **Leadership and Team Building:** Leader and Leadership, Leadership Traits, Culture and Leadership, Leadership Styles and Trends, Team Building, Types of Teams, **Decision Making and Negotiation:** Introduction to Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts **Stress and Time Management:** Stress, Sources of Stress, Ways to Cope with Stress	1) Learned about Leadership, team building, decision making and stress management. 2) Developed professional, social and academic skills to harness hidden strengths, capabilities and knowledge equip them to excel in real work environment and corporate life. 3) The students learned about various concepts like creativity, motivation, capacity building. 4) The students are able to learn leadership roles and team building. 5) The students learned the decision making techniques, negotiation styles and ways to handle stress.

Class: FYCS Semester: II

Course: **USCS201 Programming with C (Credits : 2 Lectures/Week: 3)**

Unit	Syllabus	Outcomes
I	**Structure of C program**: Header and body, Use of comments. Interpreters vs compilers, Python vs C. Compilation of a program. Formatted I/O: printf(), scanf(). **Data**: Variables, Constants, data types like: int, float char, double and void, short and long size qualifiers, signed and unsigned qualifiers. Compare with datatypes in Python. Compare static typing in C vs dynamic typing in Python **Variables**: Declaring variables, scope of the variables according to block, hierarchy of data types. Compare explicit declarations in C with implicit declarations in Python. **Types of operators**: Arithmetic, relational, logical, compound assignment, increment and decrement, conditional or ternary, bitwise and comma operators. Precedence and order of evaluation, statements and Expressions. Automatic and explicit type conversion. **Iterations**: Control statements for decision making: (i) Branching: if statement, else.. if statement, (does the writer mean if-else or nested ifs)switch statement. (ii) Looping: while loop, do.. while, for loop. (iii) Jump statements: break, continue and goto.	1. Students were able to write, compile and debug programs in C language. 2. Students were able to use different data types in a computer program. 3. Students were able to develop logics which will help them to create programs, applications in C. Also by learning the basic programming constructs they can easily switch over to any other language in future.
II	**Arrays:** (One and two dimensional), declaring array variables, initialization of arrays, accessing array elements. Compare array types of C with list and tuple types of Python. **Data Input and Output functions**: Character I/O format: getch(), getche(), getchar(), getc(), gets(), putchar(), putc(), puts(). **Manipulating Strings**: Declaring and initializing String variables, Character and string handling	1. Students were able to design programs involving decision structures, loops and functions. 2. Students were able to understand how to use functions and How to handle stings in C.

	functions. Compare with Python strings. **Functions**: Function declaration, function definition, Global and local variables, return statement, Calling a function by passing values. **Recursion**: Definition, Recursive functions.	
III	**Pointer:** Fundamentals, Pointer variables, Referencing and de-referencing, Pointer Arithmetic, Using Pointers with Arrays, Using Pointers with Strings, Array of Pointers, Pointers as function arguments, Functions returning pointers. **Dynamic Memory Allocation**: malloc(), calloc(), realloc(), free() and sizeof operator. Compare with automatic garbage collection in Python. **Structure**: Declaration of structure, reading and assignment of structure variables, Array of structures, arrays within structures, structures within structures. Compare C structures with Python tuples. **Unions**: Defining and working with unions. **File handling**: Different types of files like text and binary, Different types of functions: fopen(), fclose(), fgetc(), fputc(), fgets(), fputs(), fscanf(), fprintf(), getw(), putw(), fread(), fwrite(), fseek().	1. Students were able to explain the difference between call by value and call by reference. 2. Students got familiar with the dynamics of memory by the use of pointers. 3. Students were able to use different data structures and create/update basic data files.

Course: USCS202 Programming with Python– II

Unit	Syllabus	Outcomes
I	**Python File Input-Output:** Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problem solving applications. **Regular Expressions:** Concept of regular expression, various types of regular expressions, using match function.	1. Student learned about file management & other operations. 2 Students studied pattern search operation, required in various applications.
II	**Exception handling:** What is an exception, various keywords to handle exceptions such try, catch, except, else, finally, raise. **GUI Programming in Python (using Tkinter/wxPython/Qt)**	1. Student learned importance of error free code. 2. Student studied different types of error

	What is GUI, Advantages of GUI, Introduction to GUI library. Layout management, events and bindings, fonts, colours, drawing on canvas (line, oval, rectangle, etc.) Widgets such as : frame, label, button, checkbutton, entry, listbox, message, radiobutton, text, spinbox etc **GUI event handling**- Events and bindings, fonts, colours, drawing on canvas (line, oval, rectangle, etc.)	possibilities. 3. Also learned to handle such situations. 4 Student understood need of GUI. Got Idea of designing GUI. Learned use of various widgets.
III	**Database connectivity in Python**: Installing mysql connector, accessing connector module module, using connect, cursor, execute & close functions, reading single & multiple results of query execution, executing different types of statements, executing transactions, understanding exceptions in database connectivity. **Network connectivity**: Socket module, creating server-client programs, sending email, reading from URL	1. They learned how use of database helps in python. 2. Learned to incorporate database in python program. 3. Understood basic networking concepts. 4 Got familiar with web applications & server side programming.

Course: USCS203 Linux (Credits : 2 Lectures/Week: 3)

Unit	Syllabus	Outcomes
I	Introduction History of Linux, Philosophy, Community, Terminology, Distributions, Linux kernel vs distribution. Why learn Linux? Importance of Linux in software ecosystem: web servers, supercomputers, mobile, servers. Installation Installation methods, Hands on Installation using CD/DVD or USB drive. Linux Structure Linux Architecture, Filesystem basics, The boot process, init scripts, runlevels, shutdown process, Very basic introductions to Linux processes, Packaging methods: rpm/deb, Graphical Vs Command line.	1.students will get good working knowledge of Linux, from both a graphical and command line perspective, allowing them to easily use any Linux distribution 2. students will introduced to various tools and techniques of system administrators and linux programmers. 3. Students will be introduced to types of processes.
II	Graphical Desktop Session Management, Basic Desktop Operations, Network Management, Installing and Updating Software, Text editors: gedit, vi, vim, emacs, Graphics editors, Multimedia applications. Command Line Command line mode options, Shells, Basic Commands, General Purpose Utilities, Installing Software, User management, Environment variables, Command aliases. Linux Documentation man pages, GNU info, help command, More documentation sources File Operations Filesystem, Filesystem architecture, File types, File attributes, Working with files, Backup, compression	1. It is designed for computer students who have limited or no previous exposure to Linux. 2. students will be exposed to installation of software's in linux.
III	Security Understanding Linux Security, Uses of root, sudo command, working with passwords, Bypassing user authentication, Understanding ssh Networking Basic introduction to Networking, Network protocols: http, ftp etc., IP address, DNS, Browsers, Transferring files. ssh, telnet, ping, traceroute, route, hostname, networking GUI. Basic Shell Scripting Features and capabilities, Syntax, Constructs, Modifying files, Sed, awk command, File manipulation utilities, Dealing with large files and Text, String manipulation, Boolean expressions, File	1.Student shall be able to progress as a Developer or Linux System Administrator using the acquired skill set. 2.Students will be exposed to networking commands

	tests, Case, Debugging, Regular expressions.	

Course: USCS204 Data Structures (Credits : 2 Lectures/Week: 3)

Unit	Syllabus	Outcomes
I	**Abstract Data Types:** Introduction, The Date Abstract Data Type, Bags, Iterators. Application **Arrays:** Array Structure, Python List, Two Dimensional Arrays, Matrix Abstract Data Type, Application **Sets and Maps:** Sets-Set ADT, Selecting Data Structure, List based Implementation, Maps-Map ADT, List Based Implementation, Multi-Dimensional Arrays-Multi-Array ADT, Implementing Multiarrays, Application **Algorithm Analysis:** Complexity Analysis-Big-O Notation, Evaluating Python Code, Evaluating Python List, Amortized Cost, Evaluating Set ADT, Application **Searching and Sorting:** Searching-Linear Search, Binary Search, Sorting-Bubble, Selection and Insertion Sort, Working with Sorted Lists-Maintaining Sorted List, Maintaining sorted Lists.	1. Students explored and understood the concepts of Data Structures and its significance in programming. 2. Got holistic approach to design, use and implementation of abstract data types. 3. Understand the commonly used data structures and various forms of its implementation for different applications using Python. Learn about various approaches of searching & sorting required in day today tasks.
II	**Linked Structures**: Introduction, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, Bag ADT-Linked List Implementation. Comparing Implementations, Linked List Iterators, More Ways to Build Kinked Lists, Applications-Polynomials **Stacks**: Stack ADT, Implementing Stacks-Using Python List, Using Linked List, Stack Applications-Balanced Delimiters, Evaluating Postfix Expressions **Queues**: Queue ADT, Implementing Queue-Using Python List, Circular Array, Using List, Priority Queues- Priority Queue ADT, Bounded and unbounded Priority Queues **Advanced Linked List**: Doubly Linked Lists-Organization and Operation, Circular Linked List-Organization and Operation, Multi Lists	1. Developed Ability to program various applications using different data structure in Python. 2. Learned about dynamic memory management. 3. Learned about stack & queue which is part of various day today work as well in computer processing.
III	**Recursion:** Recursive Functions, Properties of Recursion, Its working, Recursive Applications **Hash Table:** Introduction, Hashing-Linear	1. Learned about recursion & understood its application in real life.

	Probing, Clustering, Rehashing, Separate Chaining, Hash Functions **Advanced Sorting:** Merge Sort, Quick Sort, Radix Sort, Sorting Linked List **Binary Trees:** Tree Structure, Binary Tree-Properties, Implementation and Traversals, Expression Trees, Heaps and Heapsort, Search Trees.	2. Studied tree ADT which is common structure in computers. 3. Understood use & need of such structures.

Course: USCS205 Calculus (Credits : 2 Lectures/Week: 3)

Unit	Syllabus	Outcomes
I	**DERIVATIVES AND ITS APPLICATIONS:** Review of Functions, limit of a function, continuity of a function, derivative function. Derivative In Graphing And Applications: Analysis of Functions: Increase, Decrease, Concavity, Relative Extrema; Graphing Polynomials, Rational Functions,–Cusps and Vertical Tangents. Absolute Maxima and Minima, Applied Maximum and Minimum Problems, Newton's Method.	1. Student learned important concepts of Calculus. 2. Got familiar with topics from as basic as definition of functions to partial derivatives of functions in a gradual and logical way. 3. Students solve many examples to a get compete clarity and understanding of the topics covered.
II	**INTEGRATION AND ITS APPLICATIONS:** An Overview of the Area Problem, Indefinite Integral, Definition of Area as a Limit; Sigma Notation, Definite Integral, Evaluating Definite Integrals by Substitution, Area Between Two Curves, Length of a Plane Curve. Numerical Integration: Simpson's Rule. Modeling with Differential Equations, Separation of Variables, Slope Fields, Euler's Method, First-Order Differential Equations and Applications.	1. Students developed ability to appreciate real world applications which uses these concepts.
III	**PARTIAL DERIVATIVES AND ITS APPLICATIONS:** Functions of Two or More Variables Limits and Continuity Partial Derivatives, Differentiability, Differentials, and Local Linearity, Chain Rule, Directional Derivatives and Gradients, Tangent Planes and Normal, Vectors, Maxima and Minima of Functions of Two Variables.	1. Also developed Skill to formulate a problem through Mathematical modeling and simulation.

Course: USCS206 Subject: Statistical Methods and Testing of Hypothesis
(Credits : 2 Lectures/Week: 3)

Unit	Syllabus	Outcomes
I	Standard distributions: random variable; discrete, continuous, expectation and variance of a random variable, pmf, pdf, cdf, reliability, Introduction and properties without proof for following distributions; binomial, normal, chi-square, t, F. Examples	It will focus on the random variable, mathematical expectation, and different types of distributions, sampling theory and estimation theory.
II	Hypothesis testing: one sided, two sided hypothesis, critical region, p-value, tests based on t, Normal and F, confidence intervals. Analysis of variance : one-way, two-way analysis of variance	There are two possible outcomes: if the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery
III	Non-parametric tests: need of non-parametric tests, sign test, Wilicoxon's signed rank test, run test, Kruskal-Walis tests. Post-hoc analysis of one-way analysis of variance : Duncan's test Chi-square test of association	1. Apply non-parametric techniques to statistical inference situations in which the normal-based statistics do not apply. 2. Identify and use software appropriate for nonparametric methods.

USCS207 Green Technologies Ability Enhancement Course2

Unit	Syllabus	Outcomes
I	**Green IT Overview:** Introduction , Environmental Concerns and Sustainable Development, Environmental Impacts of IT, Green I , Holistic Approach to Greening IT, Greening IT, Applying IT for Enhancing Environmental Sustainability, Green IT Standards and Eco-Labelling of IT , Enterprise Green IT Strategy, Green Washing, Green IT: Burden or Opportunity? **Green Devices and Hardware:** Introduction , Life Cycle of a Device or Hardware, Reuse, Recycle and Dispose **Green Software:** Introduction , Processor Power States , Energy-Saving Software Techniques, Evaluating and Measuring Software Impact to Platform Power **Sustainable Software Development:** Introduction, Current Practices, Sustainable Software, Software Sustainability Attributes, Software Sustainability Metrics, Sustainable Software Methodology, Defining Actions	1) Students are familiarized with the concept of Green Computing and Green IT infrastructure for making computing and information system environment sustainable. 2) Students are now aware about important concepts like 3R's of Green IT, Green Washing, Green IT strategy. 3) Students have learned various energy-saving software techniques, how to measure software impact to power, also various they are now known to various software sustainability attributes.
II	**Green Data Centres:** Data Centres and Associated Energy Challenges, Data Centre IT Infrastructure, Data Centre Facility Infrastructure: Implications for Energy Efficiency, IT Infrastructure Management, Green Data Centre Metrics **Green Data Storage:** Introduction , Storage Media Power Characteristics, Energy Management Techniques for Hard Disks, System-Level Energy Management **Green Networks and Communications:** Introduction, Objectives of Green Network Protocols, Green Network Protocols and Standards **Enterprise Green IT Strategy:** Introduction, Approaching Green IT Strategies, Business Drivers of Green IT Strategy, Business Dimensions for Green IT Transformation, Organizational	6) Students learned about that green IT can be achieved in and by hardware, software, network communication and data center operations. 7) The students have gained knowledge about optimizing software and hardware designs for development of Green IT Storage, Communication and Services. 8) The students have learned about various Green IT business drivers, business dimensions, steps in developing Green IT

	Considerations in a Green IT Strategy, Steps in Developing a Green IT Strategy, Metrics and Measurements in Green Strategies.	strategy.
III	**Sustainable Information Systems and Green Metrics:** Introduction, Multilevel Sustainable Information, Sustainability Hierarchy Models, Product Level Information, Individual Level Information, Functional Level Information, Organizational Level Information, Measuring the Maturity of Sustainable ICT **Enterprise Green IT Readiness:** Introduction, Readiness and Capability, Development of the G-Readiness Framework, Measuring an Organization's G-Readiness **Sustainable IT Services: Creating a Framework for Service Innovation:** Introduction, Factors Driving the Development of Sustainable IT, Sustainable IT Services (SITS), SITS Strategic Framework **Green Enterprises and the Role of IT:** Introduction, Organizational and Enterprise Greening, Information Systems in Greening Enterprises, Greening the Enterprise: IT Usage and Hardware, Inter-organizational Enterprise Activities and Green Issues	6) Understand the strategies, frameworks, processes and management of green IT. 7) To highlight useful approaches to embrace green IT initiatives. 8) Students have learned important concepts like Green IT Readiness Framework, SITS Strategic Framework, Greening information systems.

Class: SYCS **Semester:III**

USCS301 Theory Computation

Unit	Syllabus	Outcomes
I	Automata Theory: Defining Automaton, Finite Automaton, Transitios and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NDFA equivalence, Mealy and Moore Machines, Minimizing Automata. Formal Languges: Defining Grammar, Derivations, Languges generated by Grammar, Comsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata	1. Students understood grammar, languages and other elements of modern language design. 2. Also developed capabilities to design and develop formulations for computing models and identify its applications in diverse areas. 3. Learn about Automata theory and its application in Language Design
II	Regular Sets and Regular Grammar: Regular Grammar, Regular Expressions, Finite automata and Regular Expressions, Pumping Lemma and its Applications, Closure Properties, Regular Sets and Regular Grammar Context Free Languages: Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping Lemma for CFG Pushdown Automata: Definitions, Acceptance by PDA, PDA and CFG	1. Learned about Turing Machines and Pushdown Automata.
III	Linear Bound Automata: The Linear Bound Automata Model, Linear Bound Automata and Languages. Turing Machines: Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine, Undecidability: The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvable Problems	1. Understood Linear Bound Automata and its applications.

USCS302 Core JAVA

Unit	Syllabus	Outcomes
I	The Java Language: Features of Java, Java programming format, Java Tokens, Java Statements, Java Data Types, Typecasting, Arrays OOPS: Introduction, Class, Object, Static Keywords, Constructors, this Key Word, Inheritance, super Key Word, Polymorphism (overloading and overriding), Abstraction, Encapsulation, Abstract Classes, Interfaces String Manipulations: String, String Buffer, String Tokenizer Packages: Introduction to predefined packages (java.lang, java.util, java.io, java.sql, java.swing), User Defined Packages, Access specifiers	1. Revised Object oriented programming concepts using Java. 2. Gain knowledge of input, its processing and getting suitable output.
II	Exception Handling: Introduction, Pre-Defined Exceptions, Try-Catch-Finally, Throws, throw, User Defined Exception examples Multithreading: Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, Wait() notify() notify all() methods I/O Streams: Introduction, Byte-oriented streams, Character- oriented streams, File, Random access File, Serialization Networking: Introduction, Socket, Server socket, Client –Server Communication	1. Students exposed to Networking package. 2. Students introduced to multitasking concept. 3. Students are abled to handle errors.
III	Wrapper Classes: Introduction, Byte, Short, Integer, Long, Float, Double, Character, Boolean classes Collection Framework: Introduction, util Package interfaces, List, Set, Map, List interface & its classes, Set interface & its classes, Map interface & its classes Inner Classes: Introduction, Member inner class, Static inner class, Local inner class, Anonymous inner class AWT: Introduction, Components, Event-Delegation-Model, Listeners, Layouts, Individual components Label, Button, CheckBox, Radio Button, Choice, List, Menu, Text Field, Text Area	Gain knowledge and implementation of AWT package.

Course: USCS303 Operating System (Credits : 02 Lectures/Week:03)

Unit	Syllabus	Outcomes
I	Introduction and Operating-Systems Structures: Definition of Operating system, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments Operating-System Structures: Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure Processes: Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication. Threads: Overview, Multicore Programming, Multithreading Models	1. Students understood operating system, its structures and functioning. 2. Students understood the concepts of process. 3. Students understood about threads.
II	Process Synchronization: General structure of a typical process, race condition, 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 (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling), Thread Scheduling Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock	1. Students were able to develop and master understanding of algorithms used by operating systems for various purposes. 2. Students learned about deadlock.
III	Main Memory: Background, Logical address space, Physical address space, MMU, 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 Mass-Storage Structure: Overview, Disk Structure, Disk Scheduling, Disk Management File-System Interface: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing File-System Implementation: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management	1. Student understood the memory allocation and disk utilization of operating system. 2. Students learned about file management, disk management.

Course: USCS304 Database Management Systems (Credits : 2 Lectures/Week: 3)

Unit	Syllabus	Outcomes
I	Stored Procedures: Types and benefits of stored procedures, creating stored procedures, executing stored procedures, altering stored procedures, viewing stored procedures. Triggers: Concept of triggers, Implementing triggers – creating triggers, Insert, delete, and update triggers, nested triggers, viewing, deleting and modifying triggers, and enforcing data integrity through triggers. Sequences: creating sequences, referencing, altering and dropping a sequence. File Organization and Indexing: Cluster, Primary and secondary indexing, Index data structure: hash and Tree based indexing, Comparison of file organization: cost model, Heap files, sorted files, clustered files. Creating, dropping and maintaining indexes.	1. Students understood database management concepts, File management systems. 2 Students understood need of stored procedures & role of triggers.
II	Fundamentals of PL/SQL: Defining variables and constants, PL/SQL expressions and comparisons: Logical Operators, Boolean Expressions, CASE Expressions Handling, Null Values in Comparisons and Conditional Statements, PL/SQL Datatypes: Number Types, Character Types, Boolean Type, Datetime and Interval Types. Overview of PL/SQL Control Structures: Conditional Control: IF and CASE Statements, IF-THEN Statement, IF-THEN-ELSE Statement, IFTHEN-ELSIF Statement, CASE Statement, Iterative Control: LOOP and EXIT Statements, WHILE-LOOP, FOR-LOOP, Sequential Control: GOTO and NULL Statements	Students learned about use of PL/SQL
III	Transaction Management: ACID Properties, Serializability, Two-phase Commit Protocol, Concurrency Control, Lock Management, Lost Update Problem, Inconsistent Read Problem , Read-Write Locks, Deadlocks Handling, Two Phase Locking protocol. DCL Statements: Defining a transaction, Making Changes Permanent with COMMIT, Undoing Changes with ROLLBACK, Undoing Partial Changes with SAVEPOINT and ROLLBACK Crash Recovery: ARIES algorithm. The log based recovery, recovery related structures like transaction and dirty page table, Write-ahead log protocol, check points, recovery from a system crash, Redo and Undo phases.	Students understood concepts and implementations of transaction management and crash recovery

Course: USCS305 Combinatorics and Graph Theory(Credits : 2 Lectures/Week: 3)

Unit	Syllabus	Outcomes
I	Introduction to Combinatorics: Enumeration, Combinatorics and Graph Theory/ Number Theory/Geometry and Optimization, Sudoku Puzzles. Strings, Sets, and Binomial Coefficients: Strings- A First Look, Combinations, Combinatorial, The Ubiquitous Nature of Binomial Coefficients, The Binomial, Multinomial Coefficients. Induction: Introduction, The Positive Integers are Well Ordered, The Meaning of Statements, Binomial Coefficients Revisited, Solving Combinatorial Problems Recursively, Mathematical Induction, and Inductive Definitions Proofs by Induction. Strong induction	Appreciate beauty of combinatorics and how combinatorial problems naturally arise in many settings.
II	Graph Theory: Basic Notation and Terminology, Multigraphs: Loops and Multiple Edges, Eulerian and Hamiltonian Graphs, Graph Coloring, Planar Counting, Labeled Trees, A Digression into Complexity Theory. Applying Probability to Combinatorics, Small Ramsey Numbers, Estimating Ramsey Numbers, Applying Probability to Ramsey Theory, Ramsey's Theorem The Probabilistic Method	Students will be able to apply **theoretical** knowl edge acquired to solve realistic problems in real life.
III	Network Flows: Basic Notation and Terminology, Flows and Cuts, Augmenting Paths, The Ford-Fulkerson Labeling Algorithm, 15L A Concrete Example, Integer Solutions of Linear Programming Problems. Combinatorial Applications of Network Flows: Introduction, Matching in Bipartite Graphs, Chain partitioning, Pólya's Enumeration Theorem: Coloring the Vertices of a Square.	Apply combinatorial and graph theoretical concepts and networking to understand Computer Science concepts and apply them to solve problems

USCS306 Physical Computing and IoT Programming

Unit	Syllabus	Outcomes
I	**SoC and Raspberry Pi System on Chip**: What is System on chip? Structure of System on Chip. **SoC products**: FPGA, GPU, APU, Compute Units. ARM 8 Architecture: SoC on ARM 8. ARM 8 Architecture Introduction **Introduction to Raspberry Pi**: Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi. **Raspberry Pi Boot**: Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware.	Students understood SoC & its architecture. Students learned various SoC products in the market. Students understood ARM processor architecture. They got to know about one of the most famous SoC in the market ie. Raspberry Pi, used in a wide range of embedded applications.
II	**Programming Raspberry Pi Raspberry Pi and Linux**: About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands **Programing interfaces**: Introduction to Node.js, Python. **Raspberry Pi Interfaces**: UART, GPIO, I2C, SPI Useful **Implementations**: Cross Compilation, Pulse Width Modulation, SPI for Camera.	Students learned to work with the Raspbian OS through commands as well as GUI. Students learned various programming interfaces and hardware interfaces used for connecting to different peripheral devices. They understood the concept of cross compilation, pulse width modulation and their needs.

III	**Introduction to IoT**: What is IoT? IoT examples, Simple IoT LED Program. IoT and Protocols IoT Security: HTTP, UPnp, CoAP, MQTT, XMPP. **IoT Service as a Platform**: Clayster, Thinger.io, SenseIoT, carriots and Node RED. **IoT Security and Interoperability**: Risks, Modes of Attacks, Tools for Security and Interoperability	Students got to know about IoT and its real life applications. Students understood various security risks associated with the data communication between IoT devices and as a security measure they learned various protocols used for IoT security. Students get introduced to various IoT services available for the development of IoT applications..

USCS307 Skill Enhancement: Web Programming

Unit	Syllabus	Outcomes
I	**HTML5:** Fundamental Elements of HTML, Formatting Text in HTML, Organizing Text in HTML, Links and URLs in HTML, Tables in HTML, Images on a Web Page, Image Formats, Image Maps, Colors, FORMs in HTML, Interactive Elements, Working with Multimedia - Audio and Video File Formats, HTML elements for inserting Audio / Video on a web page **CSS:** Understanding the Syntax of CSS, CSS Selectors, Inserting CSS in an HTML Document, CSS properties to work with background of a Page, CSS properties to work with Fonts and Text Styles, CSS properties for positioning an element	Students learned how to write html files for designing webpages. They learned various html elements to display contents on a webpage. Students learned how to separate contents and style elements and how to give consistent look and feel to all the web pages of a website using CSS.
II	**JavaScript**: Using JavaScript in an HTML Document, Programming Fundamentals of JavaScript – Variables, Operators, Control Flow Statements, Popup Boxes, Functions – Defining and Invoking a Function, Defining Function arguments, Defining a Return Statement, Calling Functions with Timer, JavaScript Objects - String, RegExp, Math, Date, Browser Objects - Window, Navigator, History, Location, Document, Cookies, Document Object Model, Form Validation using JavaScript **XML:** Comparing XML with HTML, Advantages and Disadvantages of XML, Structure of an XML Document, XML Entity References, DTD, XSLT: XSLT Elements and Attributes - xsl:template, xsl:apply-templates, xsl:import, xsl:call-template, xsl:include, xsl:element, xsl:attribute, e xsl:attribute-set, xsl:value-of	Students learned to design interactive web pages using javascript. Students understood the use of XML and are able to write XML files, validate them against DTD and display xml data in specific style using XSLT.
III	**AJAX:** AJAX Web Application Model, How AJAX Works, XMLHttpRequest Object – Properties	Students understood how to update a webpage without reloading it using AJAX.

	and Methods, Handling asynchronous requests using AJAX **PHP**: Variables and Operators, Program Flow, Arrays, Working with Files and Directories, Working with Databases, Working with Cookies, Sessions and Headers **Introduction to jQuery**: Fundamentals, Selectors, methods to access HTML attributes, methods for traversing, manipulators, events, effects	Students learned how to validate user inputs, how to work with files and database using server side scripting language PHP. Students learned to apply various animations and effects to the html elements using JQuery.

Class: SYCS Semester:IV

USCS401 Fundamentals of Algorithms(Credits : 02 Lectures/Week:03)

Unit	Syllabus	Outcomes
I	Introduction to algorithm, Why to analysis algorithm, Running time analysis, How to Compare Algorithms, Rate of Growth, Commonly Used Rates of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega-Ω Notation, Theta-Θ Notation, Asymptotic Analysis, Properties of Notations, Commonly used Logarithms and Summations, Performance characteristics of algorithms, Master Theorem for Divide and Conquer, Divide and Conquer Master Theorem: Problems & Solutions, Master Theorem for Subtract and Conquer Recurrences, Method of Guessing and Confirming	Students understood basic principles of algorithm design and why algorithm analysis is important
II	Tree algorithms: What is a Tree? Glossary, Binary Trees, Types of Binary Trees, Properties of Binary Trees, Binary Tree Traversals, Generic Trees (N-ary Trees), Threaded Binary Tree Traversals, Expression Trees, Binary Search Trees (BSTs), Balanced Binary Search Trees, AVL (Adelson-Velskii and Landis) Trees Graph Algorithms: Introduction, Glossary, Applications of Graphs, Graph Representation, Graph Traversals, Topological Sort, Shortest Path Algorithms, Minimal Spanning Tree Selection Algorithms: What are Selection Algorithms? Selection by Sorting, Partition-based Selection Algorithm, Linear Selection Algorithm - Median of Medians Algorithm, Finding the K Smallest Elements in Sorted Order	Students understood how to implement algorithms in Python
III	Algorithms Design Techniques: Introduction, Classification, Classification by Implementation Method, Classification by Design Method Greedy Algorithms: Introduction, Greedy Strategy, Elements of Greedy Algorithms, Advantages and Disadvantages of	Understood how to transform new problems into algorithmic problems with efficient solutions . To understand algorithm design techniques for solving different problems

	Greedy Method, Greedy Applications, Understanding Greedy Technique Divide and Conquer Algorithms: Introduction, What is Divide and Conquer Strategy? Divide and Conquer Visualization, Understanding Divide and Conquer, Advantages of Divide and Conquer, Disadvantages of Divide and Conquer, Master Theorem, Divide and Conquer Applications Dynamic Programming: Introduction, What is Dynamic Programming Strategy? Properties of Dynamic Programming Strategy, Problems which can be solved using Dynamic Programming, Dynamic Programming Approaches, Examples of Dynamic Programming Algorithms, Understanding Dynamic Programming, Longest Common Subsequence	

USCS402 Advanced Java(Credits : 02 Lectures/Week: 03)

Unit	Syllabus	Outcomes
I	Swing: Need for swing components, Difference between AWT and swing, Components hierarchy, Panes, Swing components: Jlabel, JTextField and JPasswordField, JTextAres, JButton, JCheckBox, JRadioButton, JComboBox and JList JDBC: Introduction, JDBC Architecture, Types of Drivers, Statement, ResultSet, Read Only ResultSet, Updatable ResultSet, Forward Only ResultSet, Scrollable ResultSet, PreparedStatement, Connection Modes, SavePoint, Batch Updations, CallableStatement, BLOB & CLOB	Understood the concepts related to Java Technology
II	Servlets: Introduction, Web application Architecture, Http Protocol & Http Methods, Web Server & Web Container, Servlet Interface, GenericServlet, HttpServlet, Servlet Life Cycle, ServletConfig, ServletContext, Servlet Communication, Session Tracking Mechanisms JSP: Introduction, JSP LifeCycle, JSP Implicit Objects & Scopes, JSP Directives, JSP Scripting Elements, JSP Actions: Standard actions and customized actions	Student explore and understand use of Java Server Programming
III	Java Beans: Introduction, JavaBeans Properties, Examples Struts 2: Basic MVC Architecture, Struts 2 framework features, Struts 2 MVC pattern, Request life cycle, Examples, Configuration Files, Actions, Interceptors, Results & Result Types, Value Stack/OGNL JSON: Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java	Explored and understood use of Struts framework.

Course: USCS403 Computer Networks (Credits :02 Lectures/Week:03)

Unit	Syllabus	Outcomes
I	Introduction Network Models: Introduction to data communication, Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, Network types, Local Area Network, Wide Area Network, Switching, The Internet, Accessing the Internet, standards and administration Internet Standards. Network Models, Protocol layering, Scenarios, Principles of Protocol Layering, Logical Connections, TCP/IP Protocol Suite, Layered Architecture, Layers in 15L the TCP/IP Protocol Suite, Encapsulation and Decapsulation, Addressing, Multiplexing and Demultiplexing. Detailed introduction to Physical Layer, Detailed introduction to Data-Link Layer, Detailed introduction to Network Layer, Detailed introduction to Transport Layer, Detailed introduction to Application Layer. Data and Signals, Analog and Digital Data, Analog and Digital Signals, Sine Wave Phase, Wavelength, Time and Frequency Domains, Composite Signals, Bandwidth, Digital Signal, Bit Rate, Bit Length, Transmission of Digital Signals, Transmission Impairments, Attenuation, Distortion, Noise, Data Rate Limits, Performance, Bandwidth, Throughput, Latency (Delay)	1. Students understood the concepts of networking. 2. Students learned about various layers in detail.
II	Introduction to Physical Layer and Data-Link Layer: Digital Transmission digital-to-digital conversion, Line Coding, Line Coding Schemes, analog-to-digital conversion, Pulse Code Modulation (PCM), Transmission Modes, Parallel Transmission, Serial Transmission. Analog Transmission, digital-to-analog Conversion, Aspects of Digital-to-Analog Conversion, Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, analog-to-analog	1. Students were able to understand in-depth details about physical and data link layer. 2. Students learned how data in represented in these layers.

	Conversion, Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM), Multiplexing, Frequency-Division Multiplexing, Wavelength-Division Multiplexing, Time-Division Multiplexing. Transmission Media, Guided Media, Twisted-Pair Cable, Coaxial Cable, Fiber-Optic Cable. Switching, Three Methods of Switching , Circuit Switched Networks, Packet Switching, Introduction to Data-Link Layer, Nodes and Links, Services, Two Sub-layers, Three Types of addresses, Address Resolution Protocol (ARP). Error Detection and Correction, introduction, Types of Errors, Redundancy, Detection versus Correction,	
III	Network layer, Transport Layer Media Access Control (MAC), random access, CSMA, CSMA/CD, CSMA/CA, controlled access, Reservation, Polling, Token Passing, channelization, FDMA, TDMA, CDMA. Connecting Devices and Virtual LANs, connecting devices, Hubs, Link-Layer 15L Switches, Routers, Introduction to Network Layer, network layer services, Packetizing, Routing and Forwarding, Other Services, IPv4 addresses, Address Space, Classful Addressing. Unicast Routing, General Idea, Least-Cost Routing, Routing Algorithms, Distance-Vector Routing, Link-State Routing, Path-Vector Routing, Introduction to Transport Layer, Transport-Layer Services, Connectionless and Connection-Oriented Protocols. Transport-Layer Protocols, Service, Port Numbers, User Datagram Protocol, User Datagram, UDP Services, UDP Applications, Transmission Control Protocol, TCP Services, TCP Features, Segment.	1. This unit explained the in-depth details about the network, transport and application layers. 2.

USCS404 **Software Engineering**

Unit	Syllabus	Outcomes
I	Introduction: The Nature of Software, Software Engineering, The Software Process, Generic Process Model, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Component-Based Development, The Unified Process Phases, Agile Development- Agility, Agile Process, Extreme Programming Requirement Analysis and System Modeling: Requirements Engineering, Eliciting Requirements, SRS Validation, Components of SRS, Characteristics of SRS , Object-oriented design using the UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram.	4) Students have learned software development life cycle through framework activities and umbrella activities. 5) Students learned about various process models and agile development. 6) Students gained knowledge about requirement engineering, SRS and its characteristics, validations. 7) Understood have to construct UML project diagrams.
II	System Design: System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus The Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design Software Measurement and Metrics: Product Metrics – Measures, Metrics, and Indicators, Function-Based Metrics, Metrics for Object-Oriented Design, Operation-Oriented Metrics, User Interface Design Metrics, Metrics for Source Code, Halstead Metrics Applied to Testing, Metrics for Maintenance, Cyclomatic Complexity, Software Measurement - Size-Oriented, Function-Oriented Metrics, Metrics for Software Quality Software Project Management: Estimation in Project Planning Process –Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision, Project Scheduling - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-	9) Students learned system design, its specifications, metrics like cohesion and coupling. 10) Understood various software metrics with regard to elements like functions-based, object oriented, source code, quality, testing. 11) Learned how to calculated cyclomatic complexity metric. 12) Students have learned concepts like resource estimation, COCOMO model, project scheduling and effort distribution.

	Line Charts	
III	Risk Management - Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan Software Quality Assurance: Elements of SQA, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Six Sigma, Software Reliability, The ISO 9000 Quality Standards, Capability Maturity Model Software Testing : Verification and Validation, Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-Case Design	9) Students learned what is software risk, risk identification, risk projection, risk refinement and RMMM 10) Students know about software quality assurance and its standards, Six sigma. 11) Understood testing concepts like verification, validation. Also known about testing principles, testing levels. 12) Learned about White Box testing and Black Box testing.

Course: USCS405 Subject: Linear Algebra using Python

Unit	Syllabus	Outcomes
I	Field: Introduction to complex numbers, numbers in Python , Abstracting over fields, Playing with GF(2), Vector Space: Vectors are functions, Vector addition, Scalar-vector multiplication, Combining vector addition and scalar multiplication, Dictionary-based representations of vectors, Dot-product, Solving a triangular system of linear equations. Linear combination, Span, The geometry of sets of vectors, Vector spaces, Linear systems, homogeneous and otherwise	Appreciate the relevance of linear algebra in the field of computer science.
II	Matrix: Matrices as vectors, Transpose, Matrix-vector and vector-matrix multiplication in terms of linear combinations, Matrix-vector multiplication in terms of dot-products, Null space, Computing sparse matrix-vector product, Linear functions, Matrix-matrix multiplication, Inner product and outer product, 15L From function inverse to matrix inverse Basis: Coordinate systems, Two greedy algorithms for finding a set of generators, Minimum Spanning Forest and GF(2), Linear dependence, Basis , Unique representation, Change of basis, first look, Computational problems involving finding a basis Dimension: Dimension and rank, Direct sum, Dimension and linear functions, The annihilator	Students understand master properties of matrices including how to use them to solve linear systems of equations and how they are used in linear transformations between vector spaces
III	Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination, Finding a basis for the null space, Factoring integers, Inner Product: The inner product for vectors over the reals, Orthogonality, Orthogonalization: Projection orthogonal to multiple vectors, Projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, Orthogonal complement, Eigenvector: Modeling discrete dynamic processes, Diagonalization of the Fibonacci matrix, Eigenvalues and eigenvectors, Coordinate representation in terms of eigenvectors, The Internet worm, Existence of eigenvalues, Markov chains, Modeling a web surfer: PageRank.	Instill a computational thinking while learning linear algebra.

USCS406 .NET Technologies

Unit	Syllabus	Outcomes
I	**The .NET Framework:** .NET Languages, Common Language Runtime, .NET Class Library **C# Language Basics:** Comments, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods, Classes, Value Types and Reference Types, Namespaces and Assemblies, Inheritance, Static Members, Casting Objects, Partial Classes **ASP.NET:** Creating Websites, Anatomy of a Web Form - Page Directive, Doctype, Writing Code - Code-Behind Class, Adding Event Handlers, Anatomy of an ASP.NET Application - ASP.NET File Types, ASP.NET Web Folders, HTML Server Controls - View State, HTML Control Classes, HTML Control Events, HtmlControl Base Class, HtmlContainerControl Class, HtmlInputControl Class, Page Class, global.asax File, web.config File	Students understood the .NET framework and various category of application that they can build using .NET. Students learned various programming concepts of c# language and now they are able to develop Console applications using C#. Students learned basics of ASP.NET and various files and folders in ASP.NET project.
II	**Web Controls:** Web Control Classes, WebControl Base Class, List Controls, Table Controls, Web Control Events and AutoPostBack, Page Life Cycle **State Management:** ViewState, Cross-Page Posting, Query String, Cookies, Session State, Configuring Session State, Application State **Validation:** Validation Controls, Server-Side Validation, Client-Side Validation, HTML 5 Validation, Manual Validation, Validation with Regular Expressions **Rich Controls:** Calendar Control, AdRotator Control, MultiView Control **Themes and Master Pages:** How Themes Work, Applying a Simple Theme, Handling Theme Conflicts, Simple Master Page and Content Page,	Students learned various asp.net web controls. Students learned importance of website validation, types of validation and Validation controls in asp.net. Students learned how to manage user's state in website using various state management techniques. They learn how to give a consistent look and feel to a website using master pages and themes. Students practiced various ways to provide navigation to the user in the website.

	Connecting Master pages and Content Pages, Master Page with Multiple Content Regions, Master Pages and Relative Paths **Website Navigation**: Site Maps, URL Mapping and Routing, SiteMapPath Control, TreeView Control, Menu Control	
III	**ADO.NET**: Data Provider Model, Direct Data Access - Creating a Connection, Select Command, DataReader, Disconnected Data Access **Data Binding**: Introduction, Single-Value Data Binding, Repeated-Value Data Binding, Data Source Controls – SqlDataSource **Data Controls**: GridView, DetailsView, FormView **Working with XML**: XML Classes – XMLTextWriter, XMLTextReader **Caching**: When to Use Caching, Output Caching, Data Caching **LINQ**: Understanding LINQ, LINQ Basics, **ASP.NET AJAX**: ScriptManager, Partial Refreshes, Progress Notification, Timed Refreshes	Students learned to store data in a database using ADO.NET classes. Students learned various easy to handle data controls They understood how to read and write xml document using c#. Students understood the concept and types of caching. Students learned how to use LINQ for accessing data from various data sources. Students learned how to update webpage without refreshing it using AJAX controls.

USCS407 Skill Enhancement: Android Developer Fundamentals

Unit	Syllabus	Outcomes
I	What is Android? Obtaining the required tools, creating first android app, understanding the components of screen, adapting display orientation, action bar, Activities and Intents, Activity Lifecycle and Saving State, Basic Views: TextView, Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, and RadioGroup Views, progressBarView, AutoCompleteTextView, TimePicker View, DatePicker View, ListView View, Spinner View	Students understood the requirements of Mobile programming environment. Students. Learned about basic methods, tools and techniques for developing Apps. Learned android Activity and its lifecycle. Learned various UI components for designing android app. Learned intents for transferring data from one screen to another.
II	User Input Controls, Menus, Screen Navigation, RecyclerView, Drawables, Themes and Styles, Material design, Providing resources for adaptive layouts, AsyncTask and AsyncTaskLoader, Connecting to the Internet, Broadcast receivers, Services, Notifications, Alarm managers, Transferring data efficiently	Students learned more attractive UI elements such as FAB, Snackbar, Menus. Students learned to create interactive android app that can generate alarm, notification and popup dialogs. Students understood the need for user permissions while installing apk in mobile. Students learned to connect their app to the internet and retrieve data from internet.
III	Data - saving, retrieving, and loading: Overview to storing data, Shared preferences, SQLite primer, store data using SQLite database, ContentProviders, loaders to load and display data, Permissions, performance and security, Firebase and AdMob, Publish your app	Students understood various types of storages in android. Students learned how the data is shared between applications. Students understood the ways and steps to publish android app on playstore/appstore. They understood the use of firebase and AdMob for earning money by developing app.

Class: TYCS Semester: V

USCS502 Linux Server Administration (Credits: 03 Lectures/Week:03)

Unit	Syllabus	Outcomes
I	Introduction: Technical Summary of Linux Distributions, Managing Software Single-Host Administration: Managing Users and Groups, Booting and shutting down processes, File Systems, Core System Services, Process of configuring, compiling, Linux Kernel Networking and Security: TCP/IP for System Administrators, basic network Configuration, Linux Firewall (Netfilter), System and network security	Student will be able to develop Linux based systems and maintain.
II	Internet Services: Domain Name System (DNS), File Transfer Protocol (FTP), Apache web server, Simple Mail Transfer Protocol (SMTP), Post Office Protocol and Internet Mail Access Protocol (POP and IMAP), Secure Shell (SSH), Network Authentication, OpenLDAP Server, Samba and LDAP, Network authentication system (Kerberos), Domain Name Service (DNS), Security	Student will be able to install appropriate service on Linux server as per requirement.
III	Intranet Services: Network File System (NFS), Samba, Distributed File Systems (DFS), Network Information Service (NIS), Lightweight Directory Access Protocol (LDAP), Dynamic Host Configuration Protocol (DHCP), MySQL, LAMP Applications File Servers, Email Services, Chat Applications, Virtual Private Networking.	Student will have proficiency in Linux server administration.

Course: USCS503 Software Testing and Quality Assurance

Unit	Syllabus	Outcomes
I	Software Testing and Introduction to quality : Introduction, Nature of errors, an example for Testing, Definition of Quality , QA, QC, QM and SQA , Software Development Life Cycle, Software Quality Factors Verification and Validation : Definition of V &V , Different types of V & V Mechanisms, Concepts of Software Reviews, Inspection and Walkthrough Software Testing Techniques : Testing Fundamentals, Test Case Design, White Box Testing and its types, Black Box Testing and its types	1. Students learned Software Testing techniques. 2. Learned various software testing methods and strategies. 3. Understood a variety of software metrics, and identify defects and managing those defects for improvement in quality for given software.
II	Software Testing Strategies : Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing Software Metrics : Concept and Developing Metrics, Different types of Metrics, Complexity metrics Defect Management: Definition of Defects, Defect Management Process, Defect Reporting, Metrics Related to Defects, Using Defects for Process Improvement.	1. Students learned Software Testing techniques to understand how testing methods can be used as an effective tools in providing quality assurance concerning for software. 2. Students learned skills to design test case plan for testing software
III	Software Quality Assurance : Quality Concepts, Quality Movement, Background Issues, SQA activities, Software Reviews, Formal Technical Reviews, Formal approaches to SQA, Statistical Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, , SQA Plan , Six sigma, Informal Reviews 15L Quality Improvement : Introduction, Pareto Diagrams, Cause-effect Diagrams, Scatter Diagrams, Run charts Quality Costs : Defining Quality Costs, Types of Quality Costs, Quality Cost Measurement, Utilizing Quality Costs for Decision-Making	1. Understood importance of SQA activities, SQA strategy. 2. Learned about formal technical review report for software quality control and assurance.

Course: USCS504 Information and Network Security

Unit	Syllabus	Outcomes
I	Introduction: Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, AES (round details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers Public-Key Cryptography and RSA: Principles of Public-Key Cryptosystems, The RSA Algorithm	1. Students were able to understand the various encryption techniques. 2. Students understood why security in important and it can be achieved by using various techniques. 3. Students understood basic concepts of computer security including network security and cryptography.
II	Key Management: Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange Message Authentication and Hash Functions: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC Digital Signatures and Authentication: Digital Signatures, Authentication Protocols, Digital Signature Standard Authentication Applications: Kerberos, X.509 Authentication, Public-Key Infrastructure	1. This unit explained students the advanced encryption techniques using advanced hash functions. 2. Students understood the principles and practices of cryptographic techniques.
III	Electronic Mail Security: Pretty Good Privacy, S/MIME IP Security: Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management Web Security: Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction Intrusion: Intruders, Intrusion Techniques, Intrusion Detection Malicious Software: Viruses and Related Threats, Virus Countermeasures, DDOS Firewalls: Firewall Design Principles, Types of Firewalls	1. Students were able to understand key managements, various firewalls, security in transport layer. 2. Students understood a variety of generic security threats and vulnerabilities, and identify & analyze particular security problems for a given application. 3. Students understood various protocols for network security to protect against the threats in a network

Course: USCS506 Subject: Web Services

Unit	Syllabus	Outcomes
I	**Web services basics :** What Are Web Services? Types of Web Services Distributed computing infrastructure, overview of XML, SOAP, Building Web Services with JAX-WS, Registering and Discovering Web Services, Service Oriented Architecture, Web Services Development Life Cycle, Developing and consuming simple Web Services across platform	Students understood the evolution of web technology from static websites to dynamic and then to web services. Students learned why the XML and JSON are standards for communication over web. Students understood the roles & operations involved in a lifecycle of web service development. Students understood the use of WSDL and UDDI for web service description and discovery. Students have practically used SOAP web services freely available over internet for designing a web page and they are able to design their own SOAP based web service.
II	**The REST Architectural style** : Introducing HTTP, The core architectural elements of a RESTful system, Description and discovery of RESTful web services, Java tools and frameworks for building RESTful web services, JSON message format and tools and frameworks around JSON, Build RESTful web services with JAX-RS APIs, The Description and Discovery of RESTful Web Services, Design guidelines for building RESTful web services, Secure RESTful web services	Students learned the next generation web service technology that is REST. Students learned various elements of RESTful web service, its description and discovery. Students understood what is API. Students are able to differentiate between XML and JSON. Students are able to differentiate between SOAP and RESTful web service. They studied various tools for building RESTful API and also how to secure RESTful web services.
III	**Developing Service-Oriented Applications with WCF** : What Is Windows Communication Foundation, Fundamental Windows Communication Foundation Concepts, Windows Communication Foundation Architecture, WCF and .NET Framework Client Profile, Basic WCF Programming, WCF Feature Details. Web Service QoS	Students learned how to develop SOAP as well as RESTful web service in visual studio. Students understood fundamentals and architecture of WCF service. Students understood concept of interfaces and types of bindings in WCF web service. They understood various messaging patterns in WCF web service. Students understood quality measures of web service.

Course: USCS507 Subject: Game Programming

Unit	Syllabus	Outcomes
I	Mathematics for Computer Graphics, DirectX Kickstart: Cartesian Coordinate system: The Cartesian XY-plane, Function Graphs, Geometric Shapes, Polygonal Shapes, Areas of Shapes, Theorem of Pythagoras in 2D, Coordinates, Theorem of Pythagoras in 3D, 3D Polygons, Euler's Rule Vectors: Vector Manipulation, multiplying a Vector by a Scalar, Vector Addition and Subtraction, Position Vectors, Unit Vectors, Cartesian Vectors, Vector Multiplication, Scalar Product, Example of the Dot Product, The Dot Product in Lighting Calculations, The Dot Product in Back-Face Detection, The Vector Product, The Right-Hand Rule, deriving a Unit Normal Vector for a Triangle Areas, Calculating 2D Areas Transformations: 2D Transformations, Matrices, Homogeneous Coordinates, 3D Transformations, Change of Axes, Direction Cosines, rotating a Point about an Arbitrary Axis, Transforming Vectors, Determinants, Perspective Projection, Interpolation DirectX: Understanding GPU and GPU architectures. How they are different from CPU Architectures? Understanding how to solve by GPU?	Learner studied Graphics and gamming concepts with present working style of developers where everything remains on internet and they need to review it. Understand it, be a part of community and learn.
II	DirectX Pipeline and Programming: Introduction To DirectX 11: COM, Textures and Resources Formats, The swap chain and Page flipping, Depth Buffering, Texture Resource Views, Multisampling Theory and MS in Direct3D, Feature Levels Direct3D 11 Rendering Pipeline: Overview, Input Assembler Stage (IA), Vertex Shader Stage (VS), The Tessellation Stage (TS), Geometry Shader Stage (GS), Pixel Shader Stage (PS), Output merger Stage (OM) Understanding Meshes or Objects, Texturing, Lighting, Blending. Interpolation and Character Animation: Trigonometry: The Trigonometric Ratios, Inverse Trigonometric Ratios, Trigonometric Relationships, The Sine Rule, The Cosine Rule, Compound Angles,	The student are able to discuss and define the terms and principles of **game** design and **development**.

	Perimeter Relationships Interpolation: Linear Interpolant, Non-Linear Interpolation, Trigonometric Interpolation, Cubic Interpolation, Interpolating Vectors, Interpolating Quaternions Curves: Circle, Bezier, B-Splines Analytic Geometry: Review of Geometry, 2D Analytic Geometry, Intersection Points, Point in Triangle, and Intersection of circle with straight line.	
III	Introduction to Rendering Engines: Understanding the current market Rendering Engines. Understanding AR, VR and MR.Depth Mappers, Mobile Phones, Smart Glasses, HMD's Unity Engine: Multi-platform publishing, VR + AR: Introduction and working in Unity, 2D, Graphics, Physics, Scripting, Animation, Timeline, Multiplayer and Networking, UI, Navigation and Pathfinding, XR, Publishing. Scripting: Scripting Overview, Scripting Tools and Event Overview XR: VR, AR, MR, Conceptual Differences. SDK, Devices	Virtual reality used to enhance student learning and engagement. VR education transform the way educational content is delivered; it works on the premise of creating a virtual world — real or imagined — and allows users not only see it but also interact with it. Being immersed in what you're learning motivates you to fully understand it. It'll require less cognitive load to process the information.

Class: TYCS Semester:VI

Course: USCS601 Wireless Sensor Networks and Mobile Communication

Unit	Syllabus	Outcomes
I	Introduction: Introduction to Sensor Networks, unique constraints and challenges. Advantage of Sensor Networks, Applications of Sensor Networks, Mobile Adhoc NETworks (MANETs) and Wireless Sensor Networks, Enabling technologies for Wireless Sensor Networks. Sensor Node Hardware and Network Architecture: Single-node architecture, Hardware components & design constraints, Operating systems and execution environments, introduction to TinyOS and nesC. Network architecture, Optimization goals and figures of merit, Design principles for WSNs, Service interfaces of WSNs, Gateway concepts.	1. Students understood wireless and adhoc network, connecting different wireless devices and their compatibility. 2. Students understood how information is gathered in many different ways from these devices.
II	Medium Access Control Protocols: Fundamentals of MAC Protocols, MAC Protocols for WSNs, Sensor-MAC Case Study. 15L Routing Protocols : Data Dissemination and Gathering, Routing Challenges and Design Issues in Wireless Sensor Networks, Routing Strategies in Wireless Sensor Networks. Transport Control Protocols : Traditional Transport Control Protocols, Transport Protocol Design Issues, Examples of Existing Transport Control Protocols, Performance of Transport Control Protocols.	1. Students were able to list various applications of wireless sensor networks, describe the concepts, protocols, design, implementation and use of wireless sensor networks.
III	Introduction, Wireless Transmission and Medium Access Control: Applications, A short history of wireless communication. Wireless Transmission: Frequency for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular systems. Telecommunication, Satellite and Broadcast Systems: GSM: Mobile services, System architecture, Radio interface, Protocols, Localization And Calling, Handover, security, New data services; DECT: System architecture, Protocol architecture; ETRA, UMTS and IMT- 2000. Satellite Systems: History, Applications, Basics: GEO, LEO, MEO;	1. Students were able to implement and evaluate new ideas for solving wireless sensor network design issues

	Routing, Localization, Handover.	

Course: USCS602 Subject: Cloud Computing

Unit	Syllabus	Outcomes
I	Introduction to Cloud Computing, Characteristics and benefits of Cloud Computing, Basic concepts of Distributed Systems, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing. Elements of Parallel Computing. Elements of Distributed Computing. Technologies for Distributed Computing. Cloud Computing Architecture. The cloud reference model. Infrastructure as a service. Platform as a service. Software as a service. Types of clouds.	Students learned in-depth concepts of Cloud Computing, cloud technologies, architecture, implantations and applications. Students understood strength and limitations of cloud computing. Students are able to compare the theoretical part of course with the real life applications they use in day to day life. Students understood three types and three reference models of cloud.
II	Characteristics of Virtualized Environments. Taxonomy of Virtualization Techniques. Virtualization and Cloud Computing. Pros and Cons of Virtualization. Virtualization using KVM, Creating virtual machines, oVirt - management tool for virtualization environment. Open challenges of Cloud Computing	Students learned various types of virtualization and practically tried one of that. Students learned various types of hypervisors. Students are now able to create virtual machine on windows OS as well as Ubuntu OS. Students understood the challenges, pros and cons of virtualization as well as cloud computing.
III	Introduction to OpenStack, OpenStack test-drive, Basic OpenStack operations, OpenStack CLI and APIs, Tenant model operations, Quotas, Private cloud building blocks, Controller deployment, Networking deployment, Block Storage deployment, Compute deployment, deploying and utilizing OpenStack in production environments, Building a production environment, Application orchestration using OpenStack Heat	Students learned cloud computing reference models SAAS, PAAS and IAAS and practically tested various cloud services available over internet. Students learned various components of an open source software OpenStack which is a platform for developing their own cloud services. Students understood the complexity and co-ordination between various components of OpenStack.

USCS604 Information Retrieval (Credits: 03 Lectures/Week: 03)

Unit	Syllabus	Outcomes
I	Introduction to Information Retrieval: Introduction, History of IR, Components of IR, and Issues related to IR, Boolean retrieval, Dictionaries and tolerant retrieval.	Provided an overview of the important issues in classical and web information retrieval.
II	Link Analysis and Specialized Search: Link Analysis, hubs and authorities, Page Rank and HITS algorithms, Similarity, Hadoop & Map Reduce, Evaluation, Personalized search, Collaborative filtering and content-based recommendation of documents and products, handling "invisible" Web, Snippet generation, Summarization, Question Answering, Cross- Lingual Retrieval.	Student understood to apply information retrieval models.
III	Web Search Engine: Web search overview, web structure, the user, paid placement, search engine optimization/spam, Web size measurement, search engine optimization/spam, Web Search Architectures. XML retrieval: Basic XML concepts, Challenges in XML retrieval, A vector space model for XML retrieval, Evaluation of XML retrieval, Text-centric versus data-centric XML retrieval.	Student understood the field of information retrieval and its relationship to search engines.

Course: USCS606 Data Science

Unit	Syllabus	Outcomes
I	Introduction to Data Science: What is Data? Different kinds of data, 15L Introduction to high level programming language + Integrated Development Environment (IDE), Exploratory Data Analysis (EDA) + Data Visualization, Different types of data sources, Data Management: Data Collection, Data cleaning/extraction, Data analysis & Modeling	The students should be able to understand & comprehend the problem; and should be able to define suitable statistical method to be adopted.
II	Data Curation: Query languages and Operations to specify and transform data, Structured/schema based systems as users and acquirers of data Semi-structured systems as users and acquirers of data, Unstructured systems in the acquisition and structuring of data, Security and ethical considerations in relation to authenticating and authorizing access to data on remote systems, Software development tools, Large scale data systems, Amazon Web Services (AWS)	Students will get Advanced knowledge within Data Science, which includes data processing, data, machine learning, data extraction, statistics and typical programming languages for the area, including: Python and R. Specialized insight into data analysis. In-depth knowledge of scientific theory and methods in Data Science.
III	Statistical Modelling and Machine Learning: Introduction to model selection: Regularization, bias/variance tradeoff e.g. parsimony, AIC, BIC, Cross validation, Ridge regressions and penalized regression e.g. LASSO Data transformations: Dimension reduction, Feature extraction, Smoothing and aggregating Supervised Learning: Regression, linear models, Regression trees, Time-series Analysis, Forecasting, Classification: classification trees, Logistic regression, separating hyperplanes, k-NN Unsupervised Learning: Principal Components Analysis (PCA), k-means clustering, Hierarchical clustering, Ensemble methods	Apply knowledge about algorithms for statistical analysis, machine learning or data extraction in new areas within data science.

Course: USCS607 Ethical Hacking

Unit	Syllabus	Outcomes
I	Information Security : Attacks and Vulnerabilities Introduction to information security : Asset, Access Control, CIA, Authentication, Authorization, Risk, Threat, Vulnerability, Attack, Attack Surface, Malware, Security-Functionality-Ease of Use Triangle Types of malware :Worms, viruses, Trojans, Spyware, Rootkits Types of vulnerabilities : OWASP Top 10 : cross-site scripting (XSS), cross site request forgery (CSRF/XSRF), SQL injection, input parameter manipulation, broken authentication, sensitive information disclosure, XML External Entities, Broken access control, Security Misconfiguration, Using components with known vulnerabilities, Insufficient Logging and monitoring, OWASP Mobile Top 10, CVE Database Types of attacks and their common prevention mechanisms : Keystroke Logging, Denial of Service (DoS /DDoS), Waterhole attack, brute force, phishing and fake WAP, Eavesdropping, Man-in-the-middle, Session Hijacking, Clickjacking, Cookie Theft, URL Obfuscation, buffer overflow, DNS poisoning, ARP poisoning, Identity Theft, IoT Attacks, BOTs and BOTNETs Case-studies : Recent attacks – Yahoo, Adult Friend Finder, eBay, Equifax, WannaCry, Target Stores, Uber, JP Morgan Chase, Bad Rabbit	Understood the ethics, legality, methodologies and techniques of hacking Learned to identify security vulnerabilities and weaknesses in the target applications. .
II	Ethical Hacking – I (Introduction and pre-attack) Introduction: Black Hat vs. Gray Hat vs. White Hat (Ethical) hacking, Why is Ethical hacking needed?, How is Ethical hacking different from security auditing and digital forensics?, Signing NDA, Compliance and Regulatory concerns, Black box vs. White box vs. Black box, Vulnerability assessment and Penetration Testing. Approach : Planning - Threat Modeling, set up security verification standards, Set up security testing plan – When, which systems/apps, understanding functionality, black/gray/white, authenticated vs. unauthenticated, internal vs. external PT, Information gathering, Perform Manual and automated (Tools: WebInspect/Qualys, Nessus, Proxies, Metasploit) VA and PT, How WebInspect/Qualys tools work:	Understood the difference between Ethical hacking & security auditing , digital forensics Learned about penetration testing using various tools.

	Crawling/Spidering, requests forging, pattern matching to known vulnerability database and Analyzing results, Preparing report, Fixing security gaps following the report Enterprise strategy : Repeated PT, approval by security testing team, Continuous Application Security Testing, Phases: Reconnaissance/foot-printing/Enumeration, Phases: Scanning, Sniffing	
III	Ethical Hacking :Enterprise Security Phases : Gaining and Maintaining Access : Systems hacking – Windows and Linux – Metasploit and Kali Linux, Keylogging, Buffer Overflows, Privilege Escalation, Network hacking - ARP Poisoning, Password Cracking, WEP Vulnerabilities, MAC Spoofing, MAC Flooding, IPSpoofing, SYN Flooding, Smurf attack, Applications hacking : SMTP/Email-based attacks, VOIP vulnerabilities, Directory traversal, Input Manipulation, Brute force attack, Unsecured login mechanisms, SQL injection, XSS, Mobile apps security, Malware analysis : Netcat Trojan, wrapping definition, reverse engineering Phases : Covering your tracks : Steganography, Event Logs alteration Additional Security Mechanisms : IDS/IPS, Honeypots and evasion techniques, Secure Code Reviews (Fortify tool, OWASP Secure Coding Guidelines)	Gain knowledge of testing and exploiting systems using various tools and understood the impact of hacking in real time machines