Academic Council

Item No: _____

Devrukh Shikshan Prasarak Mandal's

Nya. TATYASAHEB ATHALYE ARTS, Ved. S.R. SAPRE COMMERCE & Vid. DADASAHEB PITRE SCIENCE COLLEGE, DEVRUKH [AUTONOMOUS]

Syllabus for F.Y. B.Sc.
Program: B.Sc.
Course: Computer Science
Credit Based Semester and Grading System with the Effect from
Academic Year 2019-20

Preamble

Computer Science has today become integral part of all industry domains as well as fields of academics and research. The industry requirements and technologies have been steadily and rapidly growing day by day. Organizations are increasingly opting for open source systems. The students too these days are thinking beyond career in the industry and aiming for research opportunities.

The overall syllabus of First Year B. Sc. (Computer Science) focuses on:

- To form basics concepts of Computer Science
- Introduce new trends and technologies to the students in gradual way
- Train the students for the challenges of IT industry
- Develop IT culture and etiquettes in students

We sincerely believe that any student taking this course will get very strong foundation and exposure to basics, advanced and emerging trends of the subject. We hope that the students' community and teachers' community will appreciate the treatment given to the courses in the syllabus.

We passionately thank all experts who shared their valuable feedbacks and suggestions in order to improvise the contents; we have sincerely attempted to incorporate each of them. We further thank Chairperson and members of Board of Studies for their confidence in us. Special thanks to Department of Computer Science and colleagues from various colleges, who volunteered or have indirectly, helped designing certain specialized courses and the syllabus as a whole.

B. Sc. Computer Science (Semester Pattern) B. Sc. First Year

COMPUTER SCIENCE– CURRICULUM

SEMESTER – I				
COURSE CODE	**COURSE TYPE**	**COURSE TITLE**	**CREDITS**	**LECTURE / WEEK**
USCST11	Core Subject	Computer Organization and Design	2	3
USCST12	Core Subject	Programming with Python- I	2	3
USCST13	Core Subject	Free and Open Source Software	2	3
USCST14	Core Subject	Database Systems	2	3
USCST15	Core Subject	Discrete Mathematics	2	3
USCST16	Core Subject	Descriptive Statistics and Introduction to Probability	2	3
USCST17	Ability Enhancement Course I	Soft Skills Development	2	3
USCSP18	Practical	USCST11 + USCST12	2	6
USCSP19	Practical	USCST13 + USCST14	2	6
USCSPIX	Practical	USCST15 + USCST16	2	6

SEMESTER - II				
COURSE CODE	**COURSE TYPE**	**COURSE TITLE**	**CREDITS**	**LECTURE / WEEK**
USCST21	Core Subject	Programming with C	2	3
USCST22	Core Subject	Programming with Python– II	2	3
USCST23	Core Subject	Linux	2	3
USCST24	Core Subject	Data Structures	2	3
USCST25	Core Subject	Computer Graphics	2	3
USCST26	Core Subject	E - Commerce	2	3
USCST27	Ability Enhancement Course II	Green Technologies	2	3
USCSP28	Practical	USCST21 + USCST22	2	6
USCSP29	Practical	USCST23 + USCST24	2	6
USCSP2X	Practical	USCST25 + USCST26	2	6

Semester I – Theory

Course: USCST11	Computer Organization and Design (Credits : 2, Lectures/Week: 3)	
Objectives: • To understand basic input output units in Computer. • To learn basic logic circuits and their functions, Sequential circuits and functions • To understand the structure and operation of modern processors and their instruction sets **Learning Outcomes:** • Student will learn about how computer systems work and underlying principles • Student will understand the basics of digital electronics needed for computers • Student will understand the basics of instruction set architecture for reduced and complex instruction sets • Student will understand the basics of processor structure and operation • Student will understand how data is transferred between the processor and I/O devices		
Unit I	**Computer Abstractions and Technology:** Basic structure and operation of a computer, functional units and their interaction. Representation of numbers and characters. **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. [1] **Sequential circuits and functions:** State diagram and state table; finite state machines and their synthesis.	**15 L**
Unit 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.	**15 L**

	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.	
Unit III	**Basic Processor Unit:** Main components of a processor: registers and register files, ALU, control unit, instruction fetch unit, and interfaces to instruction and data memories. Datapath. Instruction fetch and execute; executing arithmetic/logic, memory access and branch instructions; hardwired and microprogrammed 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.	**15 L**
Reference book: • Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012 **Text book:** • Techmax publication book **Additional References:** • Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, ARM Edition, 2011 • R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010		

Semester I – Practical

Course: USCSP18	Practical of USCST11 (Credits : 1, Lectures/Week: 3)
USCSP18	1. Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR). 2. Simplify given Boolean expression and realize it. 3. Design and verify a half/full adder 4. Design and verify half/full subtractor 5. Design a 4 bit magnitude comparator using combinational circuits. 6. Design and verify the operation of flip-flops using logic gates. 7. Verify the operation of a counter. 8. Verify the operation of a 4 bit shift register 9. Using SPIM, write and test an adding machine program that repeatedly reads in integers and adds them into a running sum. The program should stop when it gets an input that is 0, printing out the sum at that point. 10. Using SPIM, write and test a program that reads in a positive integer using the SPIM system calls. If the integer is not positive, the program should terminate with the message "Invalid Entry"; otherwise the program should print out the names of the digits of the integers, delimited by exactly one space. For example, if the user entered "528," the output would be "Five Two Eight." # Practical No. 1 to 8 can be performed using any open source simulator (like Logisim) (Download it from https://sourceforge.net/projects/circuit/) # Practical No. 9 and 10 are required to be done using SPIM. SPIM is a self-contained simulator that will run MIPS R2000/R3000 assembly language programs. # Latest version is available at https://sourceforge.net/projects/spimsimulator/

Semester I – Theory

<table>
<tr><th>Course:
USCST12</th><th>Programming with Python- I
(Credits : 2, Lectures/Week: 3)</th><th></th></tr>
<tr><td colspan="3">Objectives:
To understand basic concepts of programming concepts.
The objective of this paper is to introduce various concepts of programming to the students using Python
Learning Outcomes:
Students will be able to understand the concepts of programming before actually starting to write programs.
Students will be able to develop logic for Problem Solving.
Students will make familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc.
Students will be able to apply the problem solving skills using syntactically simple language i.e. Python (version: 3.X or higher)</td></tr>
<tr><td>Unit I</td><td>Introduction to Python: 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 value None 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</td><td>15 L</td></tr>
</table>

	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).	
Unit II	**Function:** Advantages of function, function parameters, formal parameters, actual parameters, global and local variables. **Conditional statement:** if, if-else, if-elif-else. **Iterative statements**: The range function, for statement, 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.	**15 L**
Unit III	Anonymous functions. List comprehensions. 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.	**15 L**
Reference book: • Magnus Lie Hetland, Beginning Python: From Novice to Professional, Apress • Paul Gries, et al., Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014 **Text book:** • Techmax publication book **Additional References:** • Charles Dierbach, Introduction to Computer Science using Python, Wiley, 2013 • Paul Gries , Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014 • Adesh Pandey, Programming Languages – Principles and Paradigms, Narosa, 2008 • Website : https://www.w3schools.com/python/		

Semester I – Practical

Course: USCSP18	Practical of USCST12 (Credits : 1, Lectures/Week: 3)
USCSP18	1. Installing and setting up the Python IDLE interpreter. Executing simple statements like expression statement (numeric and Boolean types), assert, assignment, delete statements; the print function for output. 2. Script and interactive modes; defining a function in the two modes; executing a script; interactively executing a statement list (semicolon-separated sequence of simple statements); the input function. 3. Programs based on lists, conditional constructs, the for statement and the range function; interactively using the built-in functions len, sum, max, min 4. Programs related to string manipulation 5. Programs based on the while statement; importing and executing built-in functions from the time, math and random modules 6. Programs using break and continue statements. 7. Programs related to dictionaries 8. Programs using list comprehensions and anonymous functions 9. Programs using the built-in methods of the string, list and dictionary classes

Semester I – Theory

Course: USCST13	Free and Open Source Software (Credits : 2, Lectures/Week: 3)	
Objectives: • To understand basic concepts free and open source software. • The objective of this paper is use of free and open source software. **Learning Outcomes:** • Upon completion of this course, students should have a good working knowledge of Open Source ecosystem, its use, impact and importance • This course shall help student to learn Open Source methodologies • Case studies with real life examples		
Unit I	**Introduction** Introduction: open Source, Free Software, Free Software vs. Open Source software, Public Domain Software, FOSS does not mean any 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.	**15 L**
Unit 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	**15 L**

	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.	
Unit III	Introduction to open source office software's like Liber Office: Writer, Calc, Impress, and Base. Introduction to Docker, Development tools, IDEs. Virtual technology.	**15 L**
Reference book: • Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill Education, 2006 • The official Ubuntu Book, 8th Edition **Text book:** • Techmax publication book **Additional References:** • The Linux Documentation Project: http://www.tldp.org/ • Docker Project Home: http://www.docker.com • Linux kernel Home: http://kernel.org • Open Source Initiative: https://opensource.org/ • Linux Documentation Project: http://www.tldp.org/ • Wikipedia: https://en.wikipedia.org/ • https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia • Github: https://help.github.com/ • The Linux Foundation: http://www.linuxfoundation.org/		

Semester I – Practical

Course: USCSP19	Practical of USCST13 (Credits : 1, Lectures/Week: 3)
USCSP19	1. Identify any Open Source software and create detailed report about it. a. Idea b. What problem does it solves? c. Licensing model d. Intent behind making it open source e. Monetization models f. Popularity g. Impact 2. Learn at least three different open source licenses and create a brief report about them. a. History of license b. Idea c. What problems does it solve? d. Detailed licensing model e. Which popular software are released under this license? f. Any popular news associated with this license? g. Popularity h. Impact 3. Contributing to Open Source a. Identify any Open Source project of your interest b. Start contributing to the project either by i. Testing ii. Reporting bugs iii. Coding iv. Helping in documentation v. Participating in discussions vi. Participating in pre-release testing programs vii. UI development. viii. Or any other important area. 4. Hands on with Open Source Software a. Identify any open source software of your interest b. Learn it from practical view-point c. Give a brief presentation about it to the class d. Sample projects: gcc, gdb, drupal, wordpress, apache web server, mysql database 5. Contributing to Wikipedia: a. Introduction to wikipedia: operating model, license, how to contribute?

	b. Create your user account on wikipedia c. Identify any topic of your choice and contribute the missing information 6. Github a. Create and publish your own open source project: Write any simple program using your choice of programming language. b. Create a repository on github and save versions of your project. You'll learn about the staging area, committing your code, branching, and merging, c. Using GitHub to Collaborate: Get practice using GitHub or other remote repositories to share your changes with others and collaborate on multi-developer projects. You'll learn how to make and review a pull request on GitHub. d. Contribute to a Live Project: Students will publish a repository containing their reflections from the course and submit a pull request. 7. Open Source Operating Systems a. Learn any open source operating system of your choice : Linux, Android, FreeBSD, Open Solaris etc. b. Learn the installation. c. Identify the unique features of the OS of your choice. 8. Virtualization: Open Source virtualization technologies: a. Install and configure any one: VirtualBox, KVM b. Create and use virtual machines 9. Containerization: a. Containerization technologies: docker, rocket, LXD b. Install and configure any containerization technology c. Create and use containers using it 10.Linux Kernel: Learn Linux kernel with respect to: a. What is Linux kernel? b. Operating model c. Licensing Model d. How development works? e. Download kernel source code. f. Compile the Kernel

Course: USCST14	Database Systems (Credits : 2, Lectures/Week: 3)	
Objectives: The objective of this course is to introduce the concept of the DBMS with respect to the relational model, to specify the functional and data requirements for a typical database application and to understand creation, manipulation and querying of data in databases. **Learning Outcomes:** • Students should be able to evaluate business information problem and find the requirements of a problem in terms of data. • Students should be able to design the database schema with the use of appropriate data types for storage of data in database. • Students should be able to create, manipulate, query and back up the databases.		
Unit 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.	**15 L**
Unit 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	**15 L**

	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, avg, sum), group by clause, having clause	
Unit 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 **Views** (creating, altering dropping, renaming and manipulating views) **DCL Statements** (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges)	**15 L**
Reference book: • Ramez Elmasri & Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education, Sixth Edition, 2010 • Ramakrishnam, Gehrke, Database Management Systems, McGraw-Hill, 2007 • Joel Murach, Murach's MySQL, Murach, 2012 **Text book:** • Techmax publication book **Additional References:** • Robert Sheldon, Geoff Moes, Begning MySQL, Wrox Press, 2005.		

Semester I – Practical

Course: USCSP19	Practical of USCST14 (Credits : 1, Lectures/Week: 3)
USCSP19	1. For given scenario • Draw E-R diagram and convert entities and relationships to table. 2. Write relational algebra queries on the tables created in Practical-1 3. Perform the following: • Viewing all databases • Creating a Database • Viewing all Tables in a Database • Creating Tables (With and Without Constraints) • Inserting/Updating/Deleting Records in a Table • Saving (Commit) and Undoing (rollback) 4. Perform the following: • Altering a Table • Dropping/Truncating/Renaming Tables • Backing up / Restoring a Database 5. Perform the following: • Simple Queries • Simple Queries with Aggregate functions • Queries with Aggregate functions (group by and having clause) 6. Queries involving • Date Functions • String Functions • Math Functions 7. Join Queries • Inner Join • Outer Join 8. Subqueries • With IN clause • With EXISTS clause 9. Views • Creating Views (with and without check option) • Dropping views • Selecting from a view 10. DCL statements • Granting and revoking permissions

Semester I – Theory

Course: USCST15	Discrete Mathematics (Credits : 2, Lectures/Week: 3)	
Objectives: • The purpose of the course is to familiarize the prospective learners with mathematical structures that are fundamentally discrete. This course introduces sets and functions, forming and solving recurrence relations and different counting principles. These concepts are useful to study or describe objects or problems in computer algorithms and programming languages **Learning Outcomes:** • Student will get overview of theory of discrete objects, starting with relations and partially ordered sets. • Study about recurrence relations, generating function and operations on them. • Give an understanding of graphs and trees, which are widely used in software. • Provide basic knowledge about models of automata theory and the corresponding formal languages.		
Unit 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 **(c) 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 solutions of nonlinear 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.	**15 L**
Unit 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: Vandermonde's Identity, Pascal triangle, Binomial theorem, Combination with indistinct objects.	**15 L**

	(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). **(c) Languages, Grammars and Machines**: Languages , regular Expression and Regular languages, Finite state Automata, grammars, Finite state machines, Gödel numbers, Turing machines.	
Unit 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	**15 L**
Reference book: • Discrete Mathematics and Its Applications, Seventh Edition by Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. (2011) • Norman L. Biggs, Discrete Mathematics, Revised Edition, Clarendon Press, Oxford 1989. • Data Structures Seymour Lipschutz, Schaum's out lines, McGraw-Hill Inc. **Text book:** • Techmax publication book **Additional References:** • Elements of Discrete Mathematics: C.L. Liu , Tata McGraw- Hill Edition . • Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education. • Discrete Mathematics: Semyour Lipschutz, Marc Lipson, Schaum's out lines, McGraw- Hill Inc. • Foundations in Discrete Mathematics: K.D. Joshi, New Age Publication, New Delhi.		

Semester I – Practical

Course: USCSP1X	Practical of USCST15 (Credits : 1, Lectures/Week: 3)
USCSP1X	11. Graphs of standard functions such as absolute value function, inverse function, logarithmic and exponential functions, flooring and ceiling functions, trigonometric functions over suitable intervals. 12. Partial ordering sets, Hasse diagram and Lattices. 13. Recurrence relation. 14. Different counting principles. 15. Finite state Automata and Finite state machines. 16. Warshall's Algorithm. 17. Shortest Path algorithms. 18. Operations on graph. 19. Breadth and Depth First search algorithms. 20. Concept of searching, inserting and deleting from binary search trees.

Semester I – Theory

Course: USCST16	Descriptive Statistics and Introduction to Probability (Credits : 2, Lectures/Week: 3)	
Objectives: • The purpose of this course is to familiarize students with basics of Statistics. This will be essential for prospective researchers and professionals to know these basics. **Learning Outcomes:** • Enable learners to know descriptive statistical concepts • Enable study of probability and statistics concept required for Computer learners		
Unit 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**
Unit 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)	**15 L**
Unit III	**Probability :** Random experiment, sample space, events types and operations of events	**15 L**

	Probability definition : classical, axiomatic, Elementary Theorems of probability (without proof) – $0 \le P(A) \le 1$, – $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ – $P(A') = 1 - P(A)$ – $P(A) \le P(B)$ if $A \subset B$ Conditional probability, 'Bayes' theorem, independence, Examples on Probability	
Reference book: • Trivedi, K.S.(2001) : Probability, Statistics, Design of Experiments and Queuing theory, with applications of Computer Science, Prentice Hall of India, New Delhi **Text book:** • Techmax publication book **Additional References:** • Ross, S.M. (2006): A First course in probability. 6th Edn Pearson • Kulkarni, M.B., Ghatpande, S.B. and Gore, S.D. (1999): common statistical tests. Satyajeet Prakashan, Pune • Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi • Gupta, S.C. and Kapoor, V.K. (1999): Applied Statistics, S. Chand and Son's, New Delhi • Montgomery, D.C. (2001): Planning and Analysis of Experiments, wiley.		

Semester I – Practical

Course: USCSP1X	Practical of USCST16 (Credits : 1, Lectures/Week: 3)
USCSP1X	1. Frequency distribution and data presentation 2. Measures of central tendency 3. Data entry using, functions, c(), scan (), Creating vectors, Mathematical Operations: ** +/-/*/ / ^ , exp, log, log10, etc, creating vector of text type, useful functions: data, frame, matrix operations, seq(), split() etc. 4. Frequency distribution using cut(), table() 5. Data presentation 6. Summary Statistics (measures of central tendency, dispersion) 7. Measures of skewness and kurtosis 8. Correlation and regression 9. Probability 10. Conditional probability

Semester I – Theory

<table>
<tr><th>Course:
USCST17</th><th>Soft Skills Development
(Credits : 2, Lectures/Week: 3)</th><th></th></tr>
<tr><td colspan="3">Objectives:
To help learners develop their soft skills and develop their personality together with their technical skills. Developing professional, social and academic skills to harness hidden strengths, capabilities and knowledge equip them to excel in real work environment and corporate life. Understand various issues in personal and profession communication and learn to overcome them
Learning Outcomes:
Student will know about various aspects of soft skills and learn ways to develop personality
Understand the importance and type of communication in personal and professional environment.
Student will get insight into much needed technical and non-technical qualities in career planning.
Learn about Leadership, team building, decision making and stress management</td></tr>
<tr><td>Unit I</td><td>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</td><td>15 L</td></tr>
<tr><td>Unit II</td><td>Academic Skills
Employment Communication: Introduction, Resume, Curriculum Vitae, Scannable Resume, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter</td><td>15 L</td></tr>
</table>

	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 Discussions, Difference between Group Discussion, Panel Discussion and Debate, Traits, Types of Group Discussions, topic based and Case based Group Discussion, Individual Traits	
Unit 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	**15 L**
Reference book: • Soft Skills: an Integrated Approach to Maximise Personality, Gajendra S. Chauhan, Sangeeta Sharma, Wiley India		

Additional References: • Personality Development and Soft Skills, Barun K. Mitra, Oxford Press • Business Communication, Shalini Kalia, Shailja Agrawal, Wiley India • Soft Skills - Enhancing Employability, M. S. Rao, I. K. International • Cornerstone: Developing Soft Skills, Sherfield, Pearson India	

Semester II – Theory

<table>
<tr><th>Course:
USCST21</th><th>Programming with C
(Credits : 2, Lectures/Week: 3)</th><th></th></tr>
<tr><td colspan="3">Objectives:
The objective of this course is to provide a comprehensive study of the C programming language, stressing upon the strengths of C, which provide the students with the means of writing modular, efficient, maintainable, and portable code.
Learning Outcomes:
Students will be able to write, compile and debug programs in C language.
Students will be able to use different data types in a computer program.
Students will be able to design programs involving decision structures, loops and functions.
Students will be able to explain the difference between call by value and call by reference
Students will be able to understand the dynamics of memory by the use of pointers.
Students will be able to use different data structures and create/update basic data files.</td></tr>
<tr><td>Unit I</td><td>Structure of C program: Header and body, Use of comments. Interpreter 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.</td><td>15 L</td></tr>
</table>

Unit 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 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.	**15 L**
Unit 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().	**15 L**
Reference book: • Programming in ANSI C (Third Edition) : E Balagurusamy, TMH **Text book:** • Techmax publication book **Additional References:** • Pradip Dey, Manas Ghosh, "Programming in C", second edition, Oxford University Press • Yashavant P. Kanetkar. " Let Us C", BPB Publications		

Semester II – Practical

Course: USCSP28	Practical of USCST21 (Credits : 1, Lectures/Week: 3)
USCSP28	1. Programs to understand the basic data types and I/O. 2. Programs on Operators and Expressions 3. Programs on decision statements. 4. Programs on looping. 5. Programs on arrays. 6. Programs on functions. 7. Programs on structures and unions. 8. Programs on pointers. 9. Programs on string manipulations. 10. Programs on basic file operations.

Semester II – Theory

<table>
<tr><th>Course:
USCST22</th><th>Programming with Python – II
(Credits : 2, Lectures/Week: 3)</th><th></th></tr>
<tr><td colspan="3">Objectives:
The objective of this paper is to explore the style of structured programming to give the idea to the students how programming can be used for designing real-life applications by reading/writing to files, GUI programming, interfacing database/networks and various other features.
Learning Outcomes:
Students will be able to understand how to read/write to files using python.
Students will be able to catch their own errors that happen during execution of programs.
Students will get an introduction to the concept of pattern matching.
Students will be made familiar with the concepts of GUI controls and designing GUI applications.
Students will be able to connect to the database to move the data to/from the application.
Students will know how to connect to computers read from URL and send email.</td></tr>
<tr><td>Unit I</td><td>Python File Input-Output: Opening and closing files, various types of file modes, reading and writing to files, manipulating directories. Iterables, iterators and their problemsolving applications.
Exception handling: What is an exception, various keywords to handle exceptions such try, catch, except, else, finally, raise.
Regular Expressions: Concept of regular expression, various types of regular expressions, using match function.</td><td>15 L</td></tr>
<tr><td>Unit II</td><td>GUI Programming in Python (using Tkinter/wxPython/Qt) What is GUI, Advantages of GUI, and Introduction to GUI library. Layout management, events and bindings, fonts, colours, drawing on canvas (line, oval, rectangle, etc.) Widgets such as : frame, label, button, check button, entry, list box, message, radio button, text, spin box etc.</td><td>15 L</td></tr>
<tr><td>Unit III</td><td>Database connectivity in Python:
Installing MySQL connector, accessing connector module, using connect, cursor, execute & close functions, reading single & multiple results of query execution, executing different types of statements, executing transactions, understanding exceptions in</td><td>15 L</td></tr>
</table>

	database connectivity. **Network connectivity:** Socket module, creating server-client programs, sending email, reading from URL	
Reference book: • Paul Gries , Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014 **Text book:** • Techmax publication book **Additional References:** • James Payne , Beginning Python: Using Python 2.6 and Python 3, Wiley India, 2010 2. • A. Lukaszewski, MySQL for Python: Database Access Made Easy, Pact Publisher, 2010		

Semester II – Practical

Course: USCSP28	Practical of USCST22 (Credits : 1, Lectures/Week: 3)
USCSP28	1. Programs to read and write files. 2. Programs with iterables and iterators. 3. Program to demonstrate exception handling. 4. Program to demonstrate the use of regular expressions. 5. Program to show draw shapes & GUI controls. 6. Program to create server-client and exchange basic information. 7. Program to send email & read contents of URL.

Semester II – Theory

<table>
<tr><th>Course:
USCST23</th><th>Linux
(Credits : 2, Lectures/Week: 3)</th><th></th></tr>
<tr><td colspan="3">Objectives:
This course introduces various tools and techniques commonly used by Linux programmers, system administrators and end users to achieve their day to day work in Linux environment. It is designed for computer students who have limited or no previous exposure to Linux.
Learning Outcomes:
Upon completion of this course, students should have a good working knowledge of Linux, from both a graphical and command line perspective, allowing them to easily use any Linux distribution.
This course shall help student to learn advanced subjects in computer science practically.
Student shall be able to progress as a Developer or Linux System Administrator using the acquired skill set.</td></tr>
<tr><td>Unit I</td><td>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, and servers.

Installation Installation methods, Hands on Installation using CD/DVD or USB drive.

Linux Structure Linux Architecture, Files ystem basics, The boot process, init scripts, run levels, shutdown process, Very basic introductions to Linux processes, Packaging methods: rpm/deb, Graphical Vs Command line.</td><td>15 L</td></tr>
<tr><td>Unit II</td><td>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</td><td>15 L</td></tr>
</table>

Unit 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 tests, Case, Debugging, Regular expressions	**15 L**
Reference book: • UNIX Concepts and Applications by Sumitabha Das. • Official Ubuntu Book, 8th Edition, by Matthew Helmke & Elizabeth K. Joseph with Jose Antonio Rey and Philips Ballew, Prentice Hall **Text book:** • Techmax publication book **Additional References:** • Linux kernel Home: http://kernel.org • Open Source Initiative: https://opensource.org/ • The Linux Foundation: http://www.linuxfoundation.org/		

Semester II – Practical

Course: USCSP29	Practical of USCST23 (Credits : 1, Lectures/Week: 3)
USCSP29	1. Linux Installation: a. Install your choice of Linux distribution e.g. Ubuntu, Fedora, Debian. b. Try different installation media like CD/DVD, USB Drive to install. c. Customize desktop environment by changing different default options like changing default background, themes, and screen savers.

2.
   a. Screen Resolution: Ascertain the current screen resolution for your desktop.
   b. Networking: Get the current networking configuration for your desktop. Are you on a wired or a wireless connection? What wireless networks are available, if any?
   c. Time Settings Change the time zone of your system to (or New York Time if you are currently in Indian time). How does the displayed time change? After noting the time change, change the time zone back to your local time zone.

3. Installing and Removing Software:
   a. Install vlc package. Verify that it runs, and then remove it.

4. Documentations:
   a. Finding Info Documentation: From the command line: bring up the info page for the grep command. Bring up the usage section.
   b. Finding man pages From the command line: Bring up the man page for the 'ls' command. Scroll down to the EXAMPLES section.
   c. Finding man pages by Topic What man pages are available that document file compression?
   d. Finding man pages by Section From the command line, bring up the man page for the printf library function. Which manual page section are library functions found?
   e. Command-Line Help List the available options for the mkdir command. How can you do this?

5. Command line operations:
   a. Install any newpackage on your system
   b. Remove the package installed
   c. Find the passwd file in / using find command
   d. Create a symbolic link to the file you found in last step
   e. Create an empty file example.txt and move it in /tmp directory using relative pathname.
   f. Delete the file moved to /tmp in previous step using absolute path.
   g. Find the location of ls, ps, bash commands.

	6. File Operations: a. Explore mounted file systems on your system. b. What are different ways of exploring mounted file systems on Linux? c. Archive and backup your home directory or work directory using tar, gzip commands. d. Use dd command to create files and explore different options to dd. e. Use diff command to create diff of two files. f. Use patch command to patch a file. And analyze the patch using diff command again. 7. Use environment: a. Which account are you logged in? How do you find out? b. Display /etc/shadow file using cat and understand the importance of shadow file. How it's different than passwd file. c. Get you current working directory. d. Explore different ways of getting command history, how to run previously executed command without typing it? e. Create alias to most commonly used commands like. 8. Linux Editors: vim/emacs a. Create, modify, search, and navigate a file in editor. b. Learn all essential commands like search, search/replace, highlight, and show line numbers. 9. Linux Security: a. Use of sudo to change user privileges to root b. Identify all operations that require sudo privileges c. Create a new user and add it to sudo configuration file. d. Set password for new user. e. Modify the expiration date for new user using password ageing. f. Delete newly added user. 10. Network: a. Get IP address of your machine using ifconfig. b. If IP is not set, then assign an IP address according to your network settings. c. Get hostname of your machine. d. Use ping to check the network connectivity to remote machines. e. Use telnet/ssh to connect to remote machines and learn the difference between the two.

	f. Troubleshooting network using traceroute, ping, route commands. 11. Shell Scripting a. Searching with grep: Search for your username in the /etc/passwd file. b. Parsing files with awk: Display in a column a unique list of all the shells used for users in /etc/passwd. Which field in /etc/passwd holds the shell (user command interpreter in the manual page)? How do you make a list of unique entries, that is, no repeated entries? c. Searching and substituting with sed: Search all instances of the user command interpreter (shell) equal to /bin/false in /etc/passwd and substitute with /bin/bash using sed. d. Exit status: write a script which does ls to a non existent file. Display an exit status of the previous command. Now create the file and again display the exit status. In each task send the ls output to /dev/null e. Working with files: Write a shell script which will ask user for a directory, create that directory and switch to it and tell the user where you are using pwd command. Now use touch to create some new files followed by displaying the filenames.

Semester II – Theory

<table>
<tr><th>Course:
USCST24</th><th>Data Structures
(Credits : 2, Lectures/Week: 3)</th><th></th></tr>
<tr><td colspan="3">Objectives:
To explore and understand the concepts of Data Structures and its significance in programming. Provide and holistic approach to design, use and implement abstract data types. Understand the commonly used data structures and various forms of its implementation for different applications using Python.
Learning Outcomes:
Learn about Data structures, its types and significance in computing
Explore about Abstract Data types and its implementation
Ability to program various applications using different data structure in Python</td></tr>
<tr><td>Unit I</td><td>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.</td><td>15 L</td></tr>
<tr><td>Unit II</td><td>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</td><td>15 L</td></tr>
</table>

	Advanced Linked List: Doubly Linked Lists-Organization and Operation, Circular Linked List-Organization and Operation, Multi Lists	
Unit III	**Recursion:** Recursive Functions, Properties of Recursion, Its working, Recursive Applications **Hash Table:** Introduction, Hashing-Linear 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	**15 L**
Reference book: • Data Structure and algorithm Using Python, Rance D. Necaise, 2016 Wiley India Edition • Data Structure and Algorithm in Python, Michael T. Goodrich, Robertom Tamassia, M. H. Goldwasser, 2016 Wiley India Edition **Text book:** • Techmax publication book **Additional References:** • Data Structure and Algorithmic Thinking with Python- Narasimha Karumanchi, 2015, Careermonk Publications • Fundamentals of Python: Data Structures, Kenneth Lambert, Delmar Cengage Learning		

Semester II – Practical

Course: USCSP29	Practical of USCST24 (Credits : 1, Lectures/Week: 3)
USCSP29	1. Implement Linear Search to find an item in a list. 2. Implement binary search to find an item in an ordered list. 3. Implement Sorting Algorithms a. Bubble sort b. Insertion sort c. Quick sort d. Merge Sort 4. Implement use of Sets and various operations on Sets. 5. Implement working of Stacks. (pop method to take the last item added off the stack and a push method to add an item to the stack) 6. Implement Program for a. Infix to Postfix conversion b. Postfix Evaluation 7. Implement the following a. A queue as a list which you add and delete items from. b. A circular queue. (The beginning items of the queue can be reused). 8. Implement Linked list and demonstrate the functionality to add and delete items in the linked list. 9. Implement Binary Tree and its traversals. 10. Recursive implementation of a. Factorial b. Fibonacci c. Tower of Hanoi

Semester II – Theory

<table>
<tr><th>Course:
USCST25</th><th>Computer Graphics
(Credits : 2, Lectures/Week: 3)</th><th></th></tr>
<tr><td colspan="3">Objectives:
To explore and understand the concepts of Data Structures and its significance in programming. Provide and holistic approach to design, use and implement abstract data types. Understand the commonly used data structures and various forms of its implementation for different applications using Python.
Learning Outcomes:
Learn about Data structures, its types and significance in computing
Explore about Abstract Data types and its implementation
Ability to program various applications using different data structure in Python</td></tr>
<tr><td>Unit I</td><td>Introduction to Computer Graphics
Introduction to Computer graphics and its applications, Elements of graphics Displays.
Scan Conversion of lines: Digital Differential Analyzer(DDA) algorithm, Bresenhams’ Line drawing algorithm
Scan Conversion of a circle: Bresenhams’ method of Circle drawing, Midpoint Circle Algorithm, Midpoint Ellipse Algorithm.
Introduction to Computer Graphics libraries in C.</td><td>15 L</td></tr>
<tr><td>Unit II</td><td>Design and Visualization
Viewing and Clipping
Introduction to Viewing and Clipping, Window to viewport mapping,
2D Clipping system: Point clipping, Inside-Outside Test, Introduction to Line Clipping- Mid-Point Subdivision Clipping Algorithm, Cohen-Sutherland Clipping algorithm.
Introduction to Polygon Clipping: Sutherland-Hodgeman Algorithm. Character Clipping
Curves and Object design
Introduction to Modelling of object primitives, Space Curve representation Cubic Splines, Bezier curves, Properties of Bezier curves, B-Spline curves, comparison of Bezier curves and B-Spline curves
Surface Generation and Object Design: Wire frame model, Surface of Revolution, Sweep surface design, Quadric Curved surfaces.</td><td>15 L</td></tr>
</table>

Unit III	**Advanced Computer Graphics** **Object Rendering** Visible and Hidden Surfaces: Introduction to hidden lines and surfaces, Image and Object space algorithm, Floating Horizon Algorithm, Painters algorithm, Z-Buffer algorithm **Object Rendering Models:** Introduction to object rendering, Illumination Model, **Shading Techniques:** Gouraund Shading, Phong Shading. Transparency effect, Introduction to shadows, Texture mapping **Animation and Virtual Reality** Animation and Virtual reality: Introduction to Computer Animation and Multimedia systems: Components of Animation system, Keyframing, Kinematics and Inverse Kinematics, Introduction to Morphing Introduction to Virtual Reality and Special Effects	**15 L**
Reference book: • Procedural elements of Computer Graphics, David F. Rogers, Tata McGraw Hill. • Computer Graphics, Donald Hearn, M P. Baker, PHI. • Computer Graphics: A programming Approach, Steven Harrington, McGraw-Hill. **Text book:** • Techmax publication book **Additional References:** • Computer Graphics: A programming Approach, Steven Harrington, McGraw-Hill. • Theory and Problems of Computer Graphics, Zhigang Xiang, Roy, plastock, Schaum's outline series, McGraw-Hill.		

Semester II – Practical

Course: USCSP2X	Practical of USCST25 (Credits : 1, Lectures/Week: 3)
USCSP2X	1. Study and use of graphics library for drawing primitive images. 2. Write a program to a line using following algorithms: (i) Digital Differential Analyzer (DDA), (ii) Bresenham's Line drawing method 3. Write a program to draw circle using the following algorithms (i) Bresenham's circle drawing method (ii) Midpoint circle drawing algorithm 4. Write a program to demonstrate the following primitive 2D transformations on a unit square or a triangle: (i) scaling in X or Y or Both directions (ii) translation in X or Y or Both directions (iii) shear transformation (iv) reflection about an axis (v) rotation transformation 5. Write a program to rotate a line about an arbitrary point (x,y). *[Use shift of origin]* *6.* Write a program to draw an origin centered 3D cube on the screen. (Use shift of origin and bring origin of coordinate at the center of the screen) 7. Implement line clipping algorithm using (i) Mid-Point Subdivision Clipping Algorithm (ii) Cohen-Sutherl andClipping algorithm 8. Write a program to generate a Bezier curve for the N input control points. (take n= 4, 5 and 6). 9. Generate cylinder as surface of revolution by rotating a line around an axis. (use delay for better visualization)

Semester II – Theory

Course: USCST26	E commerce (Credits : 2, Lectures/Week: 3)	
Objectives: • The objective of this paper is to teach basics of e-commerce, technologies, e-marketing, e-payment etc. **Learning Outcomes:** • Students will learn basics of e-commerce, technologies, e-marketing, e-payment etc.		
Unit I	History of e-commerce and Indian business context, www, advantages and disadvantages of e-commerce, e-commerce in India, various Indian case studies. Business models for e-commerce, different tyre of e-commerce, brokerage model, aggregator model, info-mediary model, community model, value chain model, manufacturer model, advertising model, subscription model, affilitate model Technologies of the www & e-security, internet client-server applications, networks and internet, url, software agents, internet service providers, html, javascript and xml, e-security, security on the internet, hacking, various security risks, e-business risk management issues, firewall.	**15 L**
Unit II	e-marketing, traditional marketing, identifying web presence goals, the browsing behavior model, online marketing, e-advertising, internet marketing trends, target markets, e-branding, marketing strategies E - Payment systems, digital token, e-cash, e-cheque, cryptography, digital signature, online financial services in India.	**15 L**
Unit III	E-customer relationship management, toolkit, life cycle, data mining in CRM, e supply chain management, supply chain, e-supply chain components, architecture. Customer effective web design, intelligent web sites, setting web sites goals, strategies for web site development , legal and ethical issues, IT Law 2000, phishing, copy right	**15 L**

Reference book: • The unofficial guide to starting an e-commerce business by Jason R.Rich, IDG books india. • E-Commerce (Pearson Custom Business Resources) by Kenneth C. Laudon • Electronic Commerce by Gary P. Schneider **Text book:** • E-commerce - An Indian Prespective by P.T. Joseph, S.J , PHI publication	

Semester II – Practical

Course: USCSP2X	Practical of USCST26 (Credits : 1, Lectures/Week: 3)
USCSP2X	1. Case study to find inspirational business examples & stories of e-commerce business in world 2. Case study : Indian e-commerce websites 3. Case study on different type of e-commerce websites. 4. Case Study : E-marketing website 5. Case Study : E- payment website 6. Case Study : E- customer website 7. Create account, manage e-commerce account and maintain security of the account. 8. Case study to find best e-commerce website on the basis of several criteria

Semester II – Theory

Course: USCST27	Green Technologies (Credits : 2, Lectures/Week: 3)	
Objectives: • To familiarize with the concept of Green Computing and Green IT infrastructure for making computing and information system environment sustainable. Encouraging optimized software and hardware designs for development of Green IT Storage, Communication and Services. To highlight useful approaches to embrace green IT initiatives. **Learning Outcomes:** • Learn about green IT can be achieved in and by hardware, software, network communication and data center operations. • Understand the strategies, frameworks, processes and management of green IT		
Unit 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	**15 L**
Unit 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	**15 L**

	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 Considerations in a Green IT Strategy, Steps in Developing a Green IT Strategy, Metrics and Measurements in Green Strategies.	
Unit 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	**15 L**
Reference book: • Harnessing Green IT: Principles and Practices, San Murugesan, G. R. Ganadharan, Wiley & IEEE. **Text book:** • Techmax publication book **Additional References:** • Green IT, Deepak Shikarpur, Vishwkarma Publications, 2014 • Green Communications: Principles, Concepts and Practice- Samdanis et al, J. Wiley • Green IT for Sustainable Business Practice: An ISEB Foundation Guide, Mark G. O'Neill, The Chartered Institute for IT, 2010		

Evaluation Scheme

I. **Internal Exam-30 Marks**
   i. **Activate participation and Regularity of a student in classroom activities – 10 marks**
   ii. **Online Test- 10 Marks**
   [Note: It will be conducted either using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment) or a test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform.]
   iii. **Seminars or Assignment or Other evaluation techniques- 10 Marks**

II. **External Evaluation-70 Marks**
   i. **Duration:** 150 Minutes / 2.5 Hours
   ii. **Theory question paper pattern**

Pattern		
Question	**Based on**	**Marks**
Q.1	Unit I, II, III	10
Q.2	Unit I	20
Q.3	Unit II	20
Q.4	Unit III	20

III. **Practical Examination**
   i. **Each core subject carries 50 Marks**
      External Evaluation
      30 Marks for performance in practical exam + **05 marks** (Journal) + **05 marks** (Viva)
      Internal Evaluation
      10 marks for Performance in Regular Practicals
   ii. Minimum 75 % practical from each core subjects are required to be completed and written in the journal
   iii. Certified Journal is compulsory for appearing at the time of Practical Exam

Program Outcomes (PO)

Our students in the computer science program should, at the time of their graduation, have:

- PO1: an ability to demonstrate computer science principles in real time computer software and hardware problems.
- PO2: an ability to identify, formulates, and review literature's to analyse and solve complex computer science problems.
- PO3: an ability to design, implement, and evaluate a computer-based system, component, process or program to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability and sustainability.
- PO4: an ability to use the techniques, skills and modern hardware and software tools necessary for computer science practices.
- PO5: the broad education necessary to understand the impact of computing in global, economic, environmental and societal context.
- PO6: an ability to understand professional, legal and ethical responsibilities as it pertains to computer science.
- PO7: an ability to function effectively as an individual, as a member or leader in diverse and multidisciplinary domains.
- PO8: an ability to effectively communicate technical information in speech, presentation and in writing.
- PO9: an ability to apply all principles and management skills in individual work and team work for project development in multidisciplinary domains.

Program Specific Outcomes (PSO)

PSO-1: Able to implement computer skill in the areas related to application development, website design, Data base, Hardware and networking.
PSO-2: Able to demonstrate basic knowledge in the areas such as, Software Engineering, Networking, Data base management, Web Technology and Operating Systems for building IT applications.
PSO-3: Able to implement algorithmic techniques by virtue of advance programming skills and Free and Open Source Software tools.
PSO-4: Design, model, program and test software systems and applications in varying domains including Networks, Embedded systems and Web technologies.

Program Educational Objectives (PEO)

- Develop within our graduates the IT proficiency and apply fruitfully the knowledge of Computer Science.
- Develop within our graduates, the ability to communicate effectively, function ethically and legally to fulfill societal needs.
- Instill commitment into graduates towards life-long learning to remain updated in the IT profession.