Shri Vile Parle Kelavani Mandal's

# MITHIBAI COLLEGE OF ARTS, CHAUHAN INSTITUTE OF SCIENCE & AMRUTBEN JIVANLAL COLLEGE OF COMMERCE AND ECONOMICS (AUTONOMOUS)

*NAAC Reaccredited 'A' grade, CGPA: 3.57,*
*Granted under RUSA, FIST-DST & Star College Scheme of DBT, Government of India,*
*Best College (2016-17), University of Mumbai*

Affiliated to the
**UNIVERSITY OF MUMBAI**

**Program: B.Sc.-Computer Science**

**F. Y. B. Sc.**

**Semester I & II**

**Choice Based Credit System (CBCS) with effect from the Academic year 2021-22**
**A.C. No:** 9
**Agenda No:** 4.9

Neelam Jain
Ashish Garande
Rohini Tarande
Zakir J. Mulani
Dr. Amol Joglekar
JAYSHREE RAVI

## PROGRAMME SPECIFIC OUTCOMES (PSO'S)

On completion of the B.Sc Computer Science, the learners should be enriched with knowledge and be able to-

**PS01:** To train the students for software development using different programming languages.
**PS02:** To develop the skills for problem solving in computing and other relevant disciplines.
**PS03:** To introduce emerging trends to the students in a gradual way.
**PS04:** To groom the students for facing the challenges in ICT industry.

### Preamble

Information and Communication Technology (ICT) 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 advancing. 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 B.Sc. Computer Science course structure therefore needed a fresh outlook and complete overhaul. A real genuine attempt has been made while designing the new syllabus for this three year graduate course. Not only does it prepare the students for a career in Software industry, it also motivates them towards further studies and research opportunities. In the first year i.e. for semester I & II, basic foundation of important skills required for software development is laid. The syllabus proposes to have four core subjects of Computer science and two core courses of Mathematics-Statistics. All core subjects are proposed to have theory as well as practical tracks. While the Computer Science courses will form fundamental skills for solving computational problems, the Mathematics & Statistics course will inculcate research oriented acumen. The syllabus design for further semesters encompasses more advanced and specialized courses of Computer Science. 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' fraternity will appreciate the treatment given to the courses in the syllabus.

Evaluation Pattern

The performance of the learner will be evaluated in two components. The first component will be a Continuous Assessment with a weightage of 25% of total marks per course. The second component will be a Semester end Examination with a weightage of 75% of the total marks per course. The allocation of marks for the Continuous Assessment and Semester end Examinations is as shown below:

**a) Details of Continuous Assessment (CA)**

25% of the total marks per course:

| Continuous Assessment | Details | Marks |
|---|---|---|
| **Component 1 (CA-1)** | Class Test/Research Paper Review/ Assignment/ Presentation/ Mini Project | 15 marks |
| **Component 2 (CA-2)** | Assignment/ presentation/mini project, etc | 10 marks |

Minimum 2 component of Continuous Assessment need to be conducted per course.

**b) Details of Semester End Examination**

75% of the total marks per course. Duration of examination will be two and half hours.

| Question Number | Description | Marks | Total Marks |
|---|---|---|---|
| Q.1 | Answer any 3 out of 4 Questions | 7 marks each | 3 x 7 = 21 marks |
| Q.2 | Answer any 3 out of 4 Questions | 7 marks each | 3 x 7 = 21 marks |
| Q.3 | Answer any 3 out of 4 Questions | 7 marks each | 3 x 7 = 21 marks |
| Q.4 | Answer any 3 out of 4 questions | 4 marks each | 3 x 4 = 12 marks |
| | | **Total Marks** | **75 marks** |

**Evaluation for practical papers**

In the Practical exams, there will be 20% assessment for the journal and laboratory work and 80% as term end component to be conducted as a semester end exam per course. For each course there will be one examiner per batch who will evaluate the practical.

Signature

**HOD**

Signature

**Approved by Vice –Principal**

Signature

**Approved by Principal**

The courses are as follows: -

| Semester – I | | |
|---|---|---|
| **Course Title** | **Credits** | **Lecture/Week** |
| Theory of Computation | 2 | 2 |
| Java Programming | 2 | 2 |
| Operating System | 2 | 2 |
| Database Management System | 2 | 2 |
| Advanced Web Technologies | 2 | 2 |
| Computer Networks | 2 | 2 |
| Combinatorics and Graph Theory | 2 | 2 |
| Computer Science Practical – VII | 3 | 6 |
| Computer Science Practical – VIII | 3 | 6 |

| Semester – II | | |
|---|---|---|
| **Course Title** | **Credits** | **Lecture/Week** |
| Fundamentals of Algorithms | 2 | 2 |
| Advanced Java | 2 | 2 |
| Physical Computing and IoT Programming | 2 | 2 |
| Android Developer Fundamentals | 2 | 2 |
| Linear Algebra with Python | 2 | 2 |
| .NET Technologies | 2 | 2 |
| Software Engineering | 2 | 2 |
| Computer Science Practical – IX | 3 | 6 |
| Computer Science Practical – X | 3 | 6 |

N.B.- (i) The duration of each theory lecture will be of 60 minutes. A course consists of 3 modules. For each module the number of hours allotted are 10. The total number of lecture hours for each course will thus be 30.

For theory component value of One Credit is equal to 15 learning hours.

(ii) There will be one practical per batch for all but one courses per semester. The duration of each practical will be of 2 hours.

For practical component the value of One Credit is equal to 30 learning hours.

**Thus in a week, a student will study 14 hours of theory and 12 hours of practical for each semester.**

| Program: Bachelor of Science(2021-22) | | | | Semester : I | |
|---|---|---|---|---|---|
| Course: Computer Organization and Design | | | | Course Code: USMACS101 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning Objectives:**

- To understand basic digital electronics.
- To understand structure and organization of computers.
- To understand the structure and operation of modern processors and their instruction sets.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** To learn about how computer systems work and underlying principles

**CO2**: To understand the basics of digital electronics needed for computers

**CO3:** To understand the basics of instruction set architecture for reduced and complex instruction sets

**CO4**: To understand the basics of processor structure and operation

**CO5**: To understand how data is transferred between the processor and I/O devices.

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Computer Abstractions and Technology, Fundamentals of Digital Logic. | 10 |
| 2 | Memory System Organization, Input /Output, Basic Processor Unit. | 10 |
| 3 | Control Unit, Instruction set architectures, Fundamentals of Advanced Computer Architecture. | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Computer Organization and Design | No. of Hours/Credits |
|---|---|---|
| 1 | **Computer Abstractions and Technology, Fundamentals of Digital Logic.** | **10** |
| | **Computer Abstractions and Technology**: Basic Computer Components and Functions. Representation of numbers and characters. | 4 |
| | **Fundamentals of Digital Logic:** Boolean Algebra, Logic Gates, Simplification of Logic Circuits: Algebraic Simplification, standard canonical form, minterm and maxterm, Karnaugh Maps. Combinational Circuits : Adders, Mux, De-Mux, Sequential Circuits : Flip-Flops (SR, JK & D), Counters : synchronous and asynchronous Counter, Register types. | 6 |
| 2 | **Memory System Organization, Input /Output, Basic Processor Unit.** | **10** |
| | **Memory System Organization:** Classification and design parameters, Memory Hierarchy, Internal Memory: RAM, SRAM and DRAM, Interleaved and Associative Memory, Cache memory, memory mapping and External Memory.<br>**Input / Output**: I/O Module, Programmed I/O, Interrupt Driven I/O, Direct Memory Access.<br>**Basic Processor Unit**: Processor Organization, Structure and Function. Register Organization, Instruction Cycle, Instruction Pipelining. | |
| 3 | **Control Unit, Instruction set architectures, Fundamentals of Advanced Computer Architecture.** | **10** |
| | **Control Unit:** Micro-Operations, Functional Requirements, Processor Control, Hardwired Implementation, Micro-programmed Control.<br>**Instruction set architectures**: Instruction Formats, Instruction Sets, Addressing Modes, word size, big-endian and little-endian | |

| | | |
|---|---|---|
| | arrangements. Instructions, sequencing. Instruction sets for RISC and CISC (examples Altera NIOS II and Freescale Cold Fire). Machine language, assembly language, assembler directives. Types of machine instructions: arithmetic, logic, shift, etc.<br>**Fundamentals of Advanced Computer Architecture: Parallel** Architecture: Classification of Parallel Systems, Flynn's Taxonomy, Array Processors, Clusters, and NUMA Computers. Introduction to Multiprocessor Systems and Multi-Core Computers. | |

| **PRACTICALS** | |
|---|---|
| **Sr. No.** | **Topic.** |
| 1 | Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR). |
| 2 | Using Logisim to design the universal gates. |
| 3 | Simplify given Boolean expression and realize it. |
| 4 | Simplify given Boolean expression using kmap. |
| 5 | Design and verify a half/full adder |
| 6 | Design and verify half/full subtractor |
| 7 | Design a 4-bit magnitude comparator using combinational circuits. |
| 8 | Design and verify the operation of flip-flops using logic gates. |
| 9 | Verify the operation of a counter. |
| 10 | Verify the operation of a 4-bit shift register |

**RECOMMENDED READING:**

**Text Books:**


1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012.
2. Computer Organization and Architecture William Stallings , 8th edition ,pearson,2012.


**Reference Books**


1. Patterson and Hennessy, Computer Organization and Design, Morgan Kaufmann, ARM Edition, 2011
2. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010

| Program: Bachelor of Science(2021-22) | | | | Semester : I | |
|---|---|---|---|---|---|
| Course: Programming with Python- I | | | | Course Code: USMACS102 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning Objectives:**

- To introduce various concepts of programming to the students using Python.
- To learn the syntax of writing various commands of Python
- To develop logic for Problem Solving with the help of Python
- To learn about the basic constructs of programming such as data, operations, conditions, loops, functions etc.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** Students should be able to understand the concepts of programming before actually starting to write programs.

**CO2:** Students should be able to develop logic reasoning skills.

**CO3:** Students should be made familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc.

**CO4:** Students should be able to apply the problem solving skills using syntactically simple language

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Introductory concepts of Python | 10 |
| 2 | Python Constructs – Conditions and Loops | 10 |
| 3 | Python dictionary and user-defined functions | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Programming with Python- I | No. of Hours/Credits |
|---|---|---|
| 1 | **Introductory concepts of Python** | **10** |
| | Introduction to Python - Working with Python Understanding Python variables Python basic Operators<br>Python Data Types Declaring and using Numeric data types: int, float, complex Using string data type and string operations Defining list and list slicing, Use of Tuple data type | |
| 2 | **Python Constructs – Conditions and Loops** | **10** |
| | Conditional blocks using if, else and elif Simple for loops in python For loop using ranges, string, list and dictionaries Use of while loops in python Loop, continue, break<br>Building blocks of python programs Understanding string in built methods List manipulation using in built methods | |
| 3 | **Python dictionary and user-defined functions** | **10** |
| | Dictionary manipulation Programming using string, list and dictionary in built functions<br>User Defined Functions - Advantages of functions, function parameters, formal parameters, actual parameters, global and local variables. Programming using functions Anonymous functions. List comprehensions. | |

| PRACTICALS | |
|---|---|
| **Sr. No.** | **Topic.** |
| 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 | Python commands based on datatypes, typecasting, built-in functions and modules. |
| 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 related to dictionaries |
| 7 | Programs using list comprehensions and anonymous functions |
| 8 | Programs using the built-in methods of the string ,list and dictionary classes. |
| 9 | Programs based on user-defined functions |

**RECOMMENDED READING:**

**Text Books:**


1. Magnus Lie Hetland, Beginning Python: From Novice to Professional, Apress
2. Paul Gries, et al., Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014


**Reference Books**


1. Charles Dierbach, Introduction to Computer Science using Python, Wiley, 2013

2. Paul Gries , Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014
3. Adesh Pandey, Programming Languages – Principles and Paradigms, Narosa, 2008

| Program: Bachelor of Science(2021-22) | | | | Semester : I | |
|---|---|---|---|---|---|
| Course: Discrete Mathematics | | | | Course Code: USMACS103 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning 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.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** To provide overview of theory of discrete objects, starting with relations and partially ordered sets.

**CO2:** Study about recurrence relations, generating function and operations on them.

**CO3:** Give an understanding of graphs and trees, which are widely helpful in software development.

**CO4:** Provide basic knowledge about models of automata theory and the corresponding formal languages.

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Recurrence Relations | 10 |
| 2 | Counting Principles, Languages and Finite State Machine | 10 |
| 3 | Graphs and Trees | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Discrete Mathematics | No. of Hours/Credits |
|---|---|---|
| 1 | **Recurrence Relations** | **10** |
| | (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.<br>(b) Relations: Definition and examples. Properties of relations , Partial Ordering sets, Linear Ordering Hasse Daigrams , Maximum and Minimum elements, Lattices<br>(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 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. . | |
| 2 | **Counting Principles, Languages and Finite State Machine** | **10** |
| | (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.<br>(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. | |
| **3** | **Graphs and Trees** | **10** |
| | 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.<br>(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 | |

| **PRACTICALS** | |
|---|---|
| **Sr. No.** | **Topic.** |
| 1 | Graphs of standard functions such as absolute value function, inverse function, logarithmic and exponential functions, flooring and ceiling functions, trigonometric functions over suitable intervals. |
| 2 | Partial ordering sets, Hasse diagram and Lattices. |
| 3 | Recurrence relation. |
| 4 | Different counting principles. |
| 5 | Finite state Automata and Finite state machines. |
| 6 | Warshall's Algorithm. |

| | |
|---|---|
| 7 | Shortest Path algorithms. |
| 8 | Operations on graph. |
| 9 | Breadth and Depth First search algorithms. |
| 10 | Concept of searching, inserting and deleting from binary search |

**RECOMMENDED READING:**

**Text Books:**


1. Discrete Mathematics and Its Applications, Seventh Edition by Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. (2011)
2. Norman L. Biggs, Discrete Mathematics, Revised Edition, Clarendon Press, Oxford 1989.
3. Data Structures Seymour Lipschutz, Schaum's out lines, McGraw- Hill Inc.


**Reference Books**


1. Elements of Discrete Mathematics: C.L. Liu , Tata McGraw- Hill Edition .
2. Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education.
3. Discrete Mathematics: Semyour Lipschutz, Marc Lipson, Schaum's out lines, McGraw- Hill Inc.
4. Foundations in Discrete Mathematics: K.D. Joshi, New Age Publication, New

| Program: Bachelor of Science(2021-22) | | | | Semester : I | |
|---|---|---|---|---|---|
| Course: Database Systems | | | | Course Code: USMACS104 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning 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.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** Evaluate business information problem and find the re*quirements of a problem in terms of data.

**CO2:** Design the database schema with the use of appropriate data types for storage of data in database.

**CO3:** To create, manipulate, query the database.

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Introduction to DBMS, Data models, Entity Relationship Model, Relational Data Model, ER to table | 10 |
| 2 | Schema refinement and Normal forms, Relational Algebra, DDL Statements, DML Statements | 10 |
| 3 | Functions, Joining Tables, Subqueries, Database Protection, Views, ,DCL Statements | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Database Systems | No. of Hours/Credits |
|---|---|---|
| 1 | **Introduction to DBMS, Data models, Entity Relationship Model, Relational Data Model, ER to table** | **10** |
| | **Introduction to DBMS** – Database, DBMS – Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture<br>**Data models** - Client/Server Architecture, Object Based Logical Model, Record Based Logical Model ( relational, hierarchical, network)<br>**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)<br>**Relational data model**– Domains, attributes, Tuples and Relations, Relational Model Notation, Characteristics of Relations, Relational Constraints - primary key, referential integrity, unique constraint, Null constraint<br>**ER to Table**- Entity to Table, Relationship to tables with and without key constraints. | |
| 2 | **Schema refinement and Normal forms, Relational Algebra, DDL Statements, DML Statements** | **10** |
| | **Schema refinement and Normal forms**: Functional dependencies, first, second, third normal form.<br>**Relational Algebra operations**: selection, projection, set operations union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division<br>**DDL Statements** - Creating Databases, Using Databases, datatypes, Creating Tables (with integrity constraints – primary key, default, check, not null), Altering Tables, Renaming Tables, Dropping Tables, | |

| | | |
|---|---|---|
| | 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 | |
| 3 | **Functions, Joining Tables, Subqueries, Database Protection, Views, ,DCL Statements** | **10** |
| | 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)<br>Joining Tables – inner join, outer join (left outer, right outer, full outer)<br>Subqueries – subqueries with IN, EXISTS, subqueries restrictions, Nested subqueries, ANY/ALL clause, correlated subqueries<br>Database Protection: Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control<br>Views (creating, altering dropping, renaming and manipulating views)<br>DCL Statements (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges) | |

| **PRACTICALS** | |
|---|---|
| **Sr. No.** | **Topic.** |
| 1 | For given scenario<br>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:<br>• Viewing all databases<br>• Creating a Database<br>• Viewing all Tables in a Database |

| | |
|---|---|
| | • Creating Tables (With and Without Constraints)<br>• Inserting/Updating/Deleting Records in a Table<br>• Saving (Commit) and Undoing (rollback) |
| 4 | Perform the following:<br>• Altering a Table<br>• Dropping/Truncating/Renaming Tables<br>• Backing up / Restoring a Database<br>• Perform the following:<br>• Simple Queries<br>• Simple Queries with Aggregate functions<br>• Queries with Aggregate functions (group by and having clause) |
| 5 | Queries involving<br>• Date Functions<br>• String Functions<br>• Math Functions |
| 6 | . Join Queries<br>• Inner Join<br>• Outer Join |
| 7 | Subqueries<br>• With IN clause<br>• With EXISTS clause |
| 8 | . Views<br>• Creating Views (with and without check option)<br>• Dropping views<br>• Selecting from a view |
| 9 | DCL statements<br>• Granting and revoking permissions |

**RECOMMENDED READING:**

**Text Books:**


1. Discrete Mathematics and Its Applications, Seventh Edition by Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. (2011)
2. Norman L. Biggs, Discrete Mathematics, Revised Edition, Clarendon Press, Oxford 1989.
3. Data Structures Seymour Lipschutz, Schaum's out lines, McGraw- Hill Inc.


**Reference Books**


1. Elements of Discrete Mathematics: C.L. Liu , Tata McGraw- Hill Edition .
2. Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education.
3. Discrete Mathematics: Semyour Lipschutz, Marc Lipson, Schaum's out lines, McGraw- Hill Inc.
4. Foundations in Discrete Mathematics: K.D. Joshi, New Age Publication, New

| Program: Bachelor of Science(2021-22) | | | | Semester : I | |
|---|---|---|---|---|---|
| Course: Algorithms & Programming in C | | | | Course Code: USMACS105 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning 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.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** Write, compile and debug programs in C language, use different data types in a computer program and design programs involving decision structures, loops and functions.

**CO2:** Explain the difference between call by value and call by reference and understand the dynamics of memory by the use of pointers.

**CO3:** Use different data structures and create/update basic data files.

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Structure of C program, Data, Variables, Types of operators and Iterations | 10 |
| 2 | Arrays, Data Input and Output functions, Manipulating Strings, Functions and Recursion | 10 |
| 3 | Pointer, Dynamic Memory Allocation and Structure. | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Algorithms & Programming in C | No. of Hours/Credits |
|---|---|---|
| 1 | **Structure of C program, Data, Variables, Types of operators and Iterations** | **10** |
| | **Structure of C program**: Header and body, Use of comments. Interpreters vs compilers, Formatted I/O: printf(), scanf().<br>**Data**: Variables, Constants, data types like: int, float char, double and void, short and long size qualifiers, signed and unsigned qualifiers.<br>**Variables**: Declaring variables, scope of the variables according to block, hierarchy of data types.<br>**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.<br>**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. | |
| 2 | **Arrays, Data Input and Output functions, Manipulating Strings, Functions and Recursion** | **10** |
| | **Arrays**: (One and two dimensional), declaring array variables, initialization of arrays, accessing array elements<br>**Data Input and Output Functions**: Character I/O format: getch(), getche(), getchar(), getc(), gets(), putchar(), putc(), puts().<br>**Manipulating Strings**: Declaring and initializing String variables, Character and string handling functions.<br>**Functions**: Function declaration, function definition, Global and local variables, return statement, Calling a function by passing values.<br>**Recursion**: Definition, Recursive functions. | |
| 3 | **Pointer, Dynamic Memory Allocation and Structure.** | **10** |

| | | |
|---|---|---|
| | **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.<br>**Dynamic Memory Allocation**: malloc(), calloc(), realloc(), free() and sizeof operator. Structure: Declaration of structure, reading and assignment of structure variables, Array of structures, arrays within structures, structures within structures. | |

| **PRACTICALS** | |
|---|---|
| **Sr. No.** | **Topic.** |
| 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 pointer arithmetic's. |
| 10 | Programs on string manipulations. |

**RECOMMENDED READING:**

**Text Books:**

1. Programming in ANSI C (Third Edition): E Balaguruswamy, TMH
2. Programming in ANSI C, RamKumar Agarwal
3. Let Us C, Yashwnt Kanitkar, BPB Publicaitions

**Reference Books:**

1. "Programming in C", Pradip Dey, 2nd Edition, Oxford University
2. C Programming Absolute Beginner's Guide (3rd Edition)' by Greg Perry
   Learn C the Hard Way' by Zed A. Shaw

| Program: Bachelor of Science(2021-22) | | | | Semester : I | |
|---|---|---|---|---|---|
| Course: Descriptive Statistics and Introduction to Probability | | | | Course Code: USMACS106 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning 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.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** Enable learners to know descriptive statistical concepts

**CO2:** Enable study of probability concept required for Computer learners

**CO3:** To groom the students for learning data analytics

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Data Presentation and Data aggregation. | 10 |
| 2 | Moments, Measures of Skewness and Kurtosis, Correlation and Regression and linear regression. | 10 |
| 3 | Probability and Probability distribution. | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Descriptive Statistics an Introduction to Probability | No. of Hours/Credits |
|---|---|---|
| 1 | **Data Presentation and Data aggregation.** | **10** |
| | **Data Presentation Data types** : attribute, variable, discrete and continuous variable Data presentation : frequency distribution, histogram o give, curves, stem and leaf display<br>**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 | |
| 2 | **Moments, Measures of Skewness and Kurtosis, Correlation and Regression and linear regression.** | **10** |
| | **Moments**: raw moments, central moments, relation between raw and central moments<br>**Measures of Skewness and Kurtosis**: based on moments, quartiles, relation between mean, median, mode for symmetric, asymmetric frequency curve.<br>**Correlation and Regression**: bivariate data, scatter plot, correlation, nonsense correlation, Karl pearson's coefficients of correlation, regression | |
| 3 | **Probability and Probability distribution.** | **10** |
| | **Probability** : Random experiment, sample space, events types and operations of events<br>**Probability definition** : classical, axiomatic, Elementary Theorems of probability (without proof)<br>- $0 \leq P(A) \leq 1$, - $P(A \ B) = P(A) + P(B) - P(A \cap B)$<br>- $P(A') = 1 - P(A)$ - $P(A) \leq P(B)$ if A B<br>Conditional probability, independence, Examples on Probability | |

| PRACTICALS | |
|---|---|
| **Sr. No.** | **Topic.** |
| 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 |

**RECOMMENDED READING:**

**Text Books:**

1. Trivedi, K.S.(2001) : Probability, Statistics, Design of Experiments and Queuing theory, with applications of Computer Science, Prentice Hall of India, New Delhi

**Reference Books:**

1. Ross, S.M. (2006): A First course in probability. 6th Edn Pearson

2. Kulkarni, M.B., Ghatpande, S.B. and Gore, S.D. (1999): common statistical tests. Satyajeet Prakashan, Pune

3. Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi

4. Gupta, S.C. and Kapoor, V.K. (1999): Applied Statistics, S. Chand and Son's, New Delhi

5. Montgomery, D.C. (2001): Planning and Analysis of Experiments, wiley.

| Program: Bachelor of Science(2021-22) | | | | Semester : I | |
|---|---|---|---|---|---|
| Course: Soft Skills Development | | | | Course Code: USMACS107 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25 ) | Term End Examinations (TEE) (Marks- 75 in Question Paper) |
| 2 | 2 | - | 2 | | |

**Learning 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:**

1) To know about various aspects of soft skills and learn ways to develop personality
2) Understand the importance and type of communication in personal and professional environment.
3) To provide insight into much needed technical and non-technical qualities in career planning.
4) Learn about Leadership, team building, decision making and stress management

**Outline of Syllabus: ( per session plan )**

| Unit | Description | Duration |
|---|---|---|
| 1 | Introduction to Soft Skills and Hard Skills Personality Development, Emotional Intelligence, Etiquette and Mannerism, Communication Today. | 10 |
| 2 | Academic Skills Employment Communication, Professional Presentation, Job Interviews, Group Discussion. | 10 |
| 3 | Professional Skills Creativity at Workplace, Ethical Values, Capacity Building: Learn, Unlearn and Relearn, Leadership and Team Building, Decision Making and Negotiation, Stress and Time Management. | 10 |

| | |
|---|---|
| **Total** | 30 |

| Module | Soft Skills Development | No. of Hours/Credits |
|---|---|---|
| 1 | **Introduction to Soft Skills and Hard Skills Personality Development, Emotional Intelligence, Etiquette and Mannerism, Communication Today.** | **10** |
| | **Introduction to Soft Skills and Hard Skills Personality Development**: Knowing Yourself, Positive Thinking, Johari's Window, Communication Skills, Non-verbal Communication, Physical Fitness<br>**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<br>**Etiquette and Mannerism**: Introduction, Professional Etiquette, Technology Etiquette<br>**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 | |
| 2 | **Academic Skills Employment Communication, Professional Presentation, Job Interviews, Group Discussion.** | **10** |
| | **Academic Skills Employment Communication:** Introduction, Resume, Curriculum Vitae, Scannable Resume, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter<br>**Professional Presentation:** Nature of Oral Presentation, Planning a Presentation, Preparing the Presentation, Delivering the Presentation<br>**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<br>**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 | |
| **3** | **Professional Skills Creativity at Workplace, Ethical Values, Capacity Building: Learn, Unlearn and Relearn, Leadership and Team Building, Decision Making and Negotiation, Stress and Time Management** | **10** |
| | **Professional Skills Creativity at Workplace:** Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at Work, The Six Thinking Hat Method<br>**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<br>**Capacity Building:** Learn, Unlearn and Relearn: Capacity Building, Elements of Capacity Building, Zones of Learning, Ideas for Learning, Strategies for Capacity Building<br>Leadership and Team Building: Leader and Leadership, Leadership Traits, Culture and Leadership, Leadership Styles and Trends, Team Building, Types of Teams,<br>**Decision Making and Negotiation:** Introduction to Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts<br>Stress and Time Management: Stress, Sources of Stress, Ways to Cope with Stress | |

**RECOMMENDED READING:**

**Textbook:**

Soft Skills: an Integrated Approach to Maximise Personality, Gajendra S. Chauhan, Sangeeta Sharma, Wiley India

**Reference Books**:

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

| Program: Bachelor of Science(2021-22) | | | | Semester : II | |
|---|---|---|---|---|---|
| Course: Principles of Web Design & Web Technologies | | | | Course Code:New Code | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning Objectives:**

- To learn how to create a user-friendly web pages
- To learn the aesthetics of web site designing
- To learn the techniques of client side scripting using JavaScript
- To be able to perform form validation in client side

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** Students should be able to understand the techniques of creating a web page.

**CO2:** Students should be able to create hyperlink between web pages

**CO3:** Students should be made familiar about the basic constructs of client side scripts using Java Script

**CO4:** Students should be able to create user forms using HTML and JavaScript for validation

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Web Site Design Principles and HTML | 10 |
| 2 | HTML CSS | 10 |
| 3 | JavaScript | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Principles of Web Design & Web Technologies | No. of Hours/Credits |
|---|---|---|
| 1 | **Web Site Design Principles and HTML** | **10** |
| | **Web Site Design Principles** – Design for the Medium, Design for the Whole Site, Design for the User, Design for the Screen Planning the Site – Create a Site Specification, Identify the Content Goal, Analyze your Audience, Build a Web Site Development Team, Filenames and URLs, Directory Structure, Diagram the Site Planning Site Navigation – Creating Usable Navigation, Using Text- Based Navigation, Using Graphics- Based Navigation Publishing and Maintaining Your Web Site – Publishing Your Web Site, Testing Your Web Site, Refining and Updating Your Content, Attracting Notice to Your Web Site<br><br>**HTML** - HTML 4.0 Tag Reference, Document Structure Tags, Formatting Tags, List Tags, Hyperlinks, Image & Image map, Table Tags, Form Tags | |
| 2 | **HTML CSS** | **10** |
| | **HTML** Frames – Understanding Frames, Frame Syntax, Targeting in Framesets, Planning Frame Content **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 Building Web Pages using HTML and CSS | |
| 3 | **Javascript** | **10** |
| | **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<br><br>JavaScript Objects - String, RegExp, Math, Date, Browser Objects - Window, Navigator, History, Location, Document, Document Object Model, Form Validation using JavaScript | |

| PRACTICALS | |
|---|---|
| **Sr. No.** | **Topic.** |
| 1 | Visit at least 5 Websites and write a report on the features, navigation and appearance aspects of them. |
| 2 | Design a webpage that makes use of<br>• Document Structure Tags<br>• Various Text Formatting Tags<br>• List Tags<br>• Image and Image Maps |
| 3 | Design a webpage that makes use of<br>• Table tags<br>• Navigation across multiple pages<br>• Embedded Multimedia elements |
| 4 | Design a webpage that makes use of Form Tags (forms with various form elements) |
| 5 | Design a webpage that make use of Cascading Style Sheets with<br>• CSS properties to change the background of a Page<br>• CSS properties to change Fonts and Text Styles<br>• CSS properties for positioning an element |
| 6 | Write JavaScript code for<br>• Performing various mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number |
| 7 | Write JavaScript code for performing Client-Side Validation of various Form Elements |
| 8 | Write JavaScript code for<br>• Demonstrating different JavaScript Objects such as String, RegExp, Math, Date<br>• Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document, c. Storing and Retrieving Cookies |

**RECOMMENDED READING:**

**Text Books:**

1. PWD: Principles of Web Design by Joel Sklar
2. HTML & CSS – The Complete Reference by Thomas Powell

**Reference Books:**

1. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press
2. Beginning javascript ,Wilton, Wrox Publication

| Program: Bachelor of Science(2021-22) | | | | Semester : II | |
|---|---|---|---|---|---|
| Course: Programming with Python – II | | | | Course Code: USMACS202 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning Objectives:**

- To learn how to create a desktop application.
- To learn the file handling and string manipulations.
- To learn the socket programming.
- To understand and manage error handling.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** Students should be able to understand how to read/write to files using python.

**CO2:** Students should be able to catch their own errors that happen during execution of programs.

**CO3:** Students should get an introduction to the concept of pattern matching.

**CO4:** Students should be made familiar with the concepts of, socket programming, GUI applications and database connectivity.

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Python File Input-Output, Exception handling and Regular Expressions. | 10 |
| 2 | GUI Programming in Python (using Tkinter/wxPython/Qt) | 10 |
| 3 | Database connectivity in Python, Object oriented python. | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Programming with Python – II | No. of Hours/Credits |
|---|---|---|
| 1 | **Python File Input-Output, Exception handling and Regular Expressions.** | **10** |
| | **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.<br>**Exception handling:** What is an exception, various keywords to handle exceptions such try, catch, except, else, finally, raise.<br>**Regular Expressions**: Concept of regular expression, various types of regular expressions, using match function. | |
| 2 | **GUI Programming in Python (using Tkinter/wxPython/Qt)** | **10** |
| | **GUI Programming in Python** (using Tkinter/wxPython/Qt) 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 | |
| 3 | **Database connectivity in Python, Object oriented python** | **10** |
| | **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. **Introduction to Object-oriented python**: python class and objects, __init__ method, inheritance. | |

| PRACTICALS | |
|---|---|
| **Sr. No.** | **Topic.** |
| 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. |
| 8 | Programs to read and write files. |
| 9 | Program for creating server and client Sockets. |
| 10 | Mini application using python. |

**RECOMMENDED READING:**

**Text Books:**


1. Paul Gries , Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014


**Reference Books:**


1. 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

| Program: Bachelor of Science(2021-22) | | | | Semester : II | |
|---|---|---|---|---|---|
| Course: Calculus | | | | Course Code: USMACS203 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning Objectives:**

The course is designed to have a grasp of important concepts of Calculus in a scientific way. It covers topics from as basic as definition of functions to partial derivatives of functions in a gradual and logical way. The learner is expected to solve as many examples as possible to a get compete clarity and understanding of the topics covered.

**Learning Outcomes:**

1) Understanding of Mathematical concepts like limit, continuity, derivative, integration of functions.
2) Ability to appreciate real world applications which uses these concepts.
3) Skill to formulate a problem through Mathematical modelling and simulation.

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Derivatives and its applications. | 10 |
| 2 | Integration and its applications. | 10 |
| 3 | Partial derivatives and its applications. | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Calculus | No. of Hours/Credit |
|---|---|---|
| 1 | **Derivatives and its applications.** | **10** |
| | **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.. | |
| 2 | **Integration and its applications.** | **10** |
| | **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. | |
| 3 | Partial derivatives and its applications. | **10** |
| | **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 | |

| PRACTICALS | |
|---|---|
| **Sr. No.** | **Topic.** |
| 1 | Continuity of functions; Derivative of functions |
| 2 | Increasing, decreasing, concave up and concave down functions |
| 3 | Relative maxima, relative minima, absolute maxima, absolute minima |
| 4 | Newton's method to find approximate solution of an equation |
| 5 | Area as a limit and length of a plane curve |
| 6 | Numerical integration using Simpson's rule |
| 7 | Solution of a first order first degree differential equation, Euler's method |
| 8 | Calculation of Partial derivatives of functions |
| 9 | Local linear approximation and directional derivatives |
| 10 | Maxima and minima of functions of two variables |

**RECOMMENDED READING:**

**Text Books:**

1. Calculus: Early transcendental (10th Edition): Howard Anton, Irl Bivens, Stephen Davis, John Wiley & sons, 2012.

**Reference Books:**

1. Calculus and analytic geometry (9th edition): George B Thomas, Ross L Finney, Addison Wesley, 1995

2. Calculus: Early Transcendentals (8th Edition): James Stewart, Brooks Cole, 2015.

3. Calculus (10th Edition): Ron Larson, Bruce H. Edwards, Cengage Learning, 2013.

| Program: Bachelor of Science(2021-22) | | | | Semester : II | |
|---|---|---|---|---|---|
| Course: Data Structures | | | | Course Code: USMACS204 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning Objectives:**

- To explore and understand the concepts of Data Structures and its significance in programming.
- To provide and holistic approach to design, use and implement abstract data types.
- To understand the commonly used data structures and various forms of its implementation for different applications using Python.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** Learn about Data structures, its types and significance in computing

**CO2:** Explore about Abstract Data types and its implementation

**CO3:** Ability to program various applications using different data structure in Python

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Abstract Data Types, Algorithm Analysis and Searching and Sorting. | 10 |
| 2 | Linked Structures, Stacks, Queues. | 10 |
| 3 | Recursion, Hash Table, Advanced Sorting and Binary Trees. | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Data Structures | No. of Hours/Credits |
|---|---|---|
| 1 | **Abstract Data Types, Algorithm Analysis and Searching and Sorting.** | **10** |
| | **Abstract Data Types**: Introduction, The Date Abstract Data Type, Bags, Iterators. Application<br>Sets and Maps: Sets-Set ADT, Selecting Data Structure, List based Implementation, Maps-Map ADT, List Based Implementation,<br>**Algorithm Analysis:** Complexity Analysis-Big-O Notation, Evaluating Python Code, Evaluating Python List, Amortized Cost, Evaluating Set ADT, Application<br>**Searching and Sorting**: Searching-Linear Search, Binary Search, Sorting-Bubble, Selection and Insertion Sort, Working with Sorted Lists-Maintaining Sorted List, Maintaining sorted Lists. | |
| 2 | **Linked Structures, Stacks, Queues** | **10** |
| | **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<br>**Stacks**: Stack ADT, Implementing Stacks-Using Python List, Using Linked List, Stack Applications-Balanced Delimiters, Evaluating Postfix Expressions<br>**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 | |

| 3 | **Recursion, Hash Table, Advanced Sorting and Binary Trees.** | **10** |
|---|---|---|
| | **Recursion**: Recursive Functions, Properties of Recursion, Its working, Recursive Applications<br>**Hash Table**: Introduction, Hashing-Linear Probing, Clustering, Rehashing, Separate Chaining, Hash Functions<br>**Advanced Sorting**: Merge Sort, Quick Sort, Radix Sort, Sorting Linked List<br>**Binary Trees**: Tree Structure, Binary Tree-Properties, Implementation and Traversals, Expression Trees, Heaps, Search Trees | |

| **PRACTICALS** | |
|---|---|
| **Sr. No.** | **Topic.** |
| 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<br>a. Bubble sort<br>b. Insertion sort<br>c. Selection 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<br>a. Infix to Postfix conversion<br>b. Postfix Evaluation<br>c. Check balanced parenthesis |
| 7 | Implement the following<br>a. A queue as a list which you add and delete items from.<br>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<br>a. Factorial<br>b. Fibonacci |

**RECOMMENDED READING:**

**Text Books:**


1) Data Structure and algorithm Using Python, Rance D. Necaise, 2016 Wiley India Edition

2) Data Structure and Algorithm in Python, Michael T. Goodrich, Robertom Tamassia, M. H. Goldwasser, 2016 Wiley India Edition


**Reference Books:**


1)Data Structure and Algorithmic Thinking with Python- Narasimha Karumanchi, 2015, Careermonk Publications

2) Fundamentals of Python: Data Structures, Kenneth Lambert, Delmar Cengage Learning

| Program: Bachelor of Science(2021-22) | | | | Semester : II | |
|---|---|---|---|---|---|
| Course: Free and Open source software with Linux | | | | Course Code: USMACS205 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning 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.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** Have a good working knowledge of Linux, from both a graphical and command line perspective, allowing them to easily use any Linux distribution.

**CO2:** Learn advanced subjects in computer science practically.

**CO3:** Progress as a Developer or Linux System Administrator using the acquired skill set.

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Linux System, shell and environment variables. | 10 |
| 2 | Linux file system, structured commands and handling user input. | 10 |
| 3 | FOSS Philosophy | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Free and Open source software with Linux | No. of Hours/Credits |
|---|---|---|
| 1 | **Linux System, shell and environment variables.** | **10** |
| | **Linux System:** Design Principles, Kernel Modules, Process Management, Scheduling, Memory Management, File system, I/O.Linux Basics: Looking into the Linux Kernel, The Unix/Linux architecture, Features of Unix/Linux.<br>**Basic bash shell commands:** Starting the shell, Shell prompt, File system Navigation, File and directory listing, File handling, Directory handling More bash shell commands: Monitoring programs, Monitoring diskspace, Working with data files: Sorting, Searching, Compressing,Archiving The **Linux environment variable**: Environment variables, setting environment variables, Removing environment variables, Default shell environment variables, setting the PATH environment variables, Locating system environment variables, Variable arrays, Using command aliases | |
| 2 | **Linux file system, structured commands and handling user input.** | **10** |
| | **Understanding Linux file permission**: Linux security, Using Linux groups, Decoding file permissions, Changing security settings Basic script building: Using multiple commands, Creating a script file, Redirecting Input and Output, Performing math<br>**Using structured commands**: Working with the if-then, if-then-else and nesting if statements, test command, Compound condition testing, advanced if then features, the case command. More structured commands: for command, C-style for command, while command, until command, nesting loops, Looping on file data, controlling the loop, processing the o/p of a loop. | |

| | | |
|---|---|---|
| | **Handling user input:** Command line parameters, Special parameter variables, shift command, Standardizing options, Getting user I/P | |
| **3** | **FOSS Philosophy** | **10** |
| | **FOSS Philosophy:** Basic Definitions, Problems with traditional commercial software, Why use Free and Open Source Software? Software Freedom, Debian Free Software Guideline, FOSS does not mean no-cost, Zero Marginal Cost in FOSS, FOSS and Ethics<br>Free Software & Open Source Software: Four essential freedoms, Free Software Definition, Free is not free. Open Source Software, Open Source Definition, Open Standards Requirement for Software<br>FOSS Development: FOSS development model, FOSS Versioning, Issue handling in FOSS, FOSS Community Development, FOSS Project Role Types, Participating in FOSS development, Starting and Maintaining a FOSS Project<br>FOSS Business Models and Licensing: Income Generation Opportunities, FOSS Business Models, Copyright, Copyleft, FOSS License, Popular FOSS licenses | |

| **PRACTICALS** | |
|---|---|
| **Sr. No.** | **Topic.** |
| 1 | i.Finding Info Documentation: From the command line: bring up the info page for the grep command. Bring up the usage section.<br>ii. Finding man pages From the command line: Bring up the man page for the 'ls' command. Scroll down to the EXAMPLES section.<br>iii. Finding man pages by Topic What man pages are available that document file compression |

| | |
|---|---|
| 2 | i. 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?<br>ii. Command line operations:<br>a. Install any newpackage on your system<br>b. Remove the package installed<br>c. Find the passwd file in / using find command<br>d. Create a symbolic link to the file you found in last step<br>e. Create an empty file example.txt and move it in /tmp directory using relative pathname.<br>f. Delete the file moved to /tmp in previous step using absolute path.<br>g. Find the location of ls, ps, bash commands |
| 3 | File Operations:<br>a. Explore mounted filesystems on your system.<br>b. What are different ways of exploring mounted filesystems on Linux?<br>c. Archive and backup your home directory or work directory using tar, gzip commands.<br>d. Use dd command to create files and explore different options to dd.<br>e. Use diff command to create diff of two files.<br>f. Use patch command to patch a file. And analyze the patch using diff command again. |
| 4 | Use environment<br>a. Which account are you logged in? How do you find out?<br>b. Display /etc/shadow file using cat and understand the importance of shadow file. How it's different than passwd file.<br>c. Get you current working directory.<br>d. Explore different ways of getting command history, how to run previously executed command without typing it?<br>e. Create alias to most commonly used commands like. |
| 5 | Linux Editors: vim/emacs<br>a. Create,modify, search, navigate a file in editor.<br>b. Learn all essential commands like search, search/replace, highlight, show line numbers. |

| | |
|---|---|
| 6 | Linux Security:<br>a. Use of sudo to change user privileges to root<br>b. Identify all operations that require sudo privileges<br>c. Create a new user and add it to sudo configuration file.<br>d. Set password for new user.<br>e. Modify the expiration date for new user using password ageing.<br>f. Delete newly added user |
| 7 | i. Searching with grep: Search for your username in the /etc/passwd file.<br>ii. 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?<br>iii. 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.<br>iv. 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<br>v. 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.<br>vi. Environment variables: Write a script which displays all environment variables on the system |
| 8 | i. Functions: Write a script that asks user for a number (1,2 or 3) which is used to call a function with the number in its name. The function then displays a message with the function number within it, example: "This message is from function number 4."<br>ii. Arithmetic: Write a script which will work as arithmetic calculator to add, subtract, multiply, divide. The user should pass an argument on the command line a letter (a,s,m or d) and two numbers. If wrong number of arguments are passed then display an error message. Make use of functions to perform operations. |

| | |
|---|---|
| | iii. Case Statements: Write a script that will be given a month number as the argument and will translate this number into a month name. The result will be printed to stdout. |

**RECOMMENDED READING:**

**Text Books:**

1) Unix Concepts and Applications by Sumitabha Das.

2) Official Ubuntu Book, 8th Edition, by Matthew Helmke & Elizabeth K. Joseph with Jose Antonio Rey and Philips Ballew, Prentice Hall

**Reference Books:**

1) Linux kernel Home: http://kernel.org

2) Open Source Initiative: https://opensource.org/

3) The Linux Foundation: http://www.linuxfoundation.org/

| Program: Bachelor of Science(2021-22) | | | | Semester : II | |
|---|---|---|---|---|---|
| Course: Statistical Methods and Testing of Hypothesis | | | | Course Code: USMACS206 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning 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.

**Course Outcomes:**

After completion of the course, learners would be able to:

**CO1:** Know descriptive statistical concepts

**CO2:** Study probability concept required for Computer learners

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Standard distributions. | 10 |
| 2 | Hypothesis testing. | 10 |
| 3 | Non-parametric tests. | 10 |
| | **Total** | **30** |
| **PRACTICALS** | | **30** |

| Module | Statistical Methods and Testing of Hypothesis | No. of Hours/Credits |
|---|---|---|
| 1 | **Standard distributions** | **10** |
| | **Standard distributions**: random variable; discrete, continuous, expectation and variance of a random variable, pmf, pdf, cdf, Introduction and properties without proof for following distributions; binomial, normal, chi-square, t, F. Examples | |
| 2 | **Hypothesis testing** | **10** |
| | **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 | |
| 3 | **Non-parametric tests.** | **10** |
| | **Non-parametric tests:** need of non-parametric tests, sign test, Wilicoxon's signed rank test, run test, Kruskal-Walis tests. Chi-square test of association | |

| **PRACTICALS** | |
|---|---|
| **Sr. No.** | **Topic.** |
| 1 | Problems based on binomial distribution |
| 2 | Problems based on normal distribution |
| 3 | Property plotting of binomial distribution |
| 4 | Property plotting of normal distribution |
| 5 | Plotting pdf, cdf, pmf, for discrete and continuous distribution |
| 6 | t test, normal test, F test |

| | |
|---|---|
| 7 | Analysis of Variance |
| 8 | Non parametric tests- I |
| 9 | Non- Parametric tests – II |
| 10 | Post-hoc analysis of one-way analysis |

**RECOMMENDED READING:**

**Text Books:**


1. Trivedi, K.S.(2009) : Probability, Statistics, Design of Experiments and Queuing theory, with applications of Computer Science, Prentice Hall of India, New Delhi


**Reference Books:**


1. Ross, S.M. (2006): A First course in probability. 6th Edn Pearson

2. Kulkarni, M.B., Ghatpande, S.B. and Gore, S.D. (1999): Common statistical tests. Satyajeet Prakashan, Pune

3. Gupta, S.C. and Kapoor, V.K. (2002) : Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi

4. Gupta, S.C. and Kapoor, V.K. (4th Edition) : Applied Statistics, S. Chand and Son's, New Delhi

5. Montgomery, D.C. (2001): Planning and Analysis of Experiments, Wiley

| Program: Bachelor of Science | | | | Semester : II | |
|---|---|---|---|---|---|
| Course: Environmental Studies | | | | Course Code: USMACS207 | |
| Teaching Scheme | | | | Evaluation Scheme | |
| Lecture (Hours per week) | Practical (Hours per week) | Tutorial (Hours per week) | Credit | Continuous Assessment and Evaluation (CAE) (Marks - 25) | Term End Examinations (TEE) (Marks-75 in Question Paper) |
| 2 | 2 | - | 2+1 | 25 | 75 |

**Learning Objectives:**

Environmental Studies (EVS) at the primary stage envisages exposing students to the real situations in their surroundings also to help students to connect and be aware of, appreciate and be sensitized towards the prevailing environmental issues (natural, physical, social and cultural).

**Learning Outcomes:**

**CO1:** To expose students to the real- life world, natural and social, in which they live.

**CO2:** To enable them to analyse, evaluate, and draw inferences about problems and concerns related to the environment.

**CO3:** To add, wherever possible the understanding of environmental issues.

**Outline of Syllabus: (per session plan)**

| Module | Description | No of hours |
|---|---|---|
| 1 | Introduction to Environmental Studies and Ecosystems, Natural Resources: Renewable and Non-Renewable Resources | 10 |
| 2 | Biodiversity and Conservation , Human Communities and Environment. | 10 |
| 3 | Environmental Pollution ,Environmental Policies and Practices | 10 |
| | **Total** | **30** |

| Module | Statistical Methods and Testing of Hypothesis | No. of Hours/Credits |
|---|---|---|
| 1 | **Introduction to Environmental Studies and Ecosystems, Natural Resources: Renewable and Non-Renewable Resources** | **10** |
| | **Introduction to Environmental Studies and Ecosystems**<br><br>Multidisciplinary nature of environmental studies; components of environment – atmosphere, hydrosphere, lithosphere and biosphere. Scope and importance; Concept of sustainability and sustainable development.<br><br>What is an ecosystem? Structure and function of ecosystem; Energy flow in an ecosystem: food chain, food web and ecological succession. Ecosystems: a) Forest ecosystem b) Grassland ecosystem c) Desert ecosystem d) Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)<br><br>**Natural Resources**: Renewable and Non-Renewable Resources Land Resources and land use change; Land degradation, soil erosion and desertification.<br><br>Deforestation: Causes and impacts due to mining, dam building on environment, forests, biodiversity and tribal populations. Water: Use and over-exploitation of surface and ground water, floods, droughts, conflicts over water (international & inter-state).<br><br>Heating of earth and circulation of air; air mass formation and precipitation. Energy resources: Renewable and non-renewable energy sources, use of alternate energy sources, growing energy needs | |

| | | |
|---|---|---|
| 2 | **Biodiversity and Conservation , Human Communities and Environment.** | **10** |
| | **Biodiversity and Conservation**<br><br>Levels of biological diversity: genetic, species and ecosystem diversity; Biogeography zones of India; Biodiversity patterns and global biodiversity hot spots India as a mega-biodiversity nation; Endangered and endemic species of India Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts, biological invasions; Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity. Ecosystem and biodiversity services: Ecological, economic, social, ethical, aesthetic and Informational value.<br><br>**Human Communities and Environment**<br><br>Human population and growth: Impacts on environment, human health and welfares. Carbon foot-print. esettlement and rehabilitation of project affected persons; case studies.<br><br>Disaster management: floods, earthquakes, cyclones and landslides. Environmental movements: Chipko, Silent valley, Bishnois of Rajasthan.Environmental ethics: Role of Indian and other religions and cultures in environmental conservation. Environmental communication and public awareness | |
| 3 | **Environmental Pollution ,Environmental Policies and Practices** | **10** |
| | **Environmental Pollution**<br><br>Environmental pollution: types, causes, effects and controls; Air, water, soil, chemical and noise pollution Nuclear hazards and human health risks Solid waste management: Control measures of urban and industrial waste. | |

| | | |
|---|---|---|
| | **Environmental Policies and Practices**<br><br>Climate change, global warming, ozone layer depletion, acid rain and impacts on human communities and agriculture.<br><br>Environment Laws: Environment Protection Act; Air (Prevention & Control of Pollution) Act; Water (Prevention and control of Pollution) Act; Wildlife Protection Act; Forest Conservation Act; International agreements; Montreal and Kyoto protocols and conservation on Biological Diversity (CBD). The Chemical Weapons Convention (CWC).Nature reserves, tribal population and rights, and human, wildlife conflicts in Indian context | |

**RECOMMENDED READING:**

**Textbook:**

1 Dr.(Smt.).Bala Krishnamoorthy, Environment Management, Text and Cases, Prentice Hall of India, 2nd Edition, 2008

**Reference Books:**

1. Agarwal S.K, Environmental Issues and Themes, A.P.H. Publishing Corporation, 1997 (Classic)
2. Dodds Felix, Earth summit 2002: A new deal by, Routledge, 2001
3. Journal of Down earth published by center for science and Education CSE.