



CIRCULAR NO.SU/B.Sc./CBC&GS /69/2023

It is hereby inform to all concerned that, the syllabi prepared by the Board of Studies, Ad-hoc Boards and recommended by the Dean, Faculty of Science & Technology, the Hon'ble Vice-Chancellor has accepted the **following syllabi of Bachelor of Science with Practical Pattern of Question Paper under the scheme of Choice Based Credit & Grading System** in his emergency powers under section 12(7) of the Maharashtra Public Universities Act, 2016 on behalf of the Academic Council as appended herewith.

Sr.No.	Courses	Semester
1.	B.Sc. Biotechnology (Optional)	IIIrd & IVth semester
2.	B.Sc. Microbiology (Optional)	IIIrd & IVth semester
3.	B.Sc. Information Technology (Optional)	IIIrd & IVth semester
4.	Bachelor of Computer Application (Optional)	IIIrd & IVth semester
5.	B.Sc.Polymer Chemistry (Optional)	IIIrd & IVth semester
6.	B.Sc.Computer Science (Degree)	IIIrd & IVth semester
7.	Honors Degree of Computer Science	IIIrd & IVth semester
8.	Honors Degree of Biotechnology	IIIrd & IVth semester

This is effective from the Academic Year 2023-24 and onwards.

All concerned are requested to note the contents of this circular and bring the notice to the students, teachers and staff for their information and necessary action.

University Campus,
Aurangabad-431 004.

REF.NO.SU/2023/ 1241-49

Date:- 12.06.2023.

★
★
★
★


Deputy Registrar,
Academic Section

Copy forwarded with compliments to :-

- 1] **The Principal of all concerned Colleges,**
Dr. Babasaheb Ambedkar Marathwada University,
- 2] **The Director, University Network & Information Centre, UNIC, with a request to upload this Circular on University Website.**

Copy to :-

- 1] **The Director, Board of Examinations & Evaluation, Dr.BAMU,A'bad.**
- 2] The Section Officer,[B.Sc.Unit] Examination Branch,Dr.BAMU,A'bad.
- 3] The Programmer [Computer Unit-1] Examinations, Dr.BAMU,A'bad.
- 4] The Programmer [Computer Unit-2] Examinations, Dr.BAMU,A'bad.
- 5] The In-charge,[E-Suvidha Kendra], Rajarshi Shahu Maharaj Pariksha Bhavan, Dr.BAMU,A'bad.
- 6] The Public Relation Officer, Dr.BAMU,A'bad.
- 7] The Record Keeper, Dr.BAMU,A'bad.

Dr. Babasaheb Ambedkar Marathwada University
Aurangabad- 431004(MS) India



**Three Year Undergraduate Bachelor Degree Program
In Science and Technology**

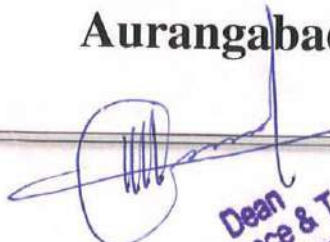
B. Sc. (Computer Science)

**Curriculum Structure and Scheme of
Examination**

Choice Based Credit System

(Effective from Academic Year 2022-23)

Dr. Babasaheb Ambedkar Marathwada University
Aurangabad – 431004 (MS) India


Dean
Faculty of Science & Technology
Dr. Babasaheb Ambedkar Marathwada
University, Aurangabad

Important Note Regarding Skill Enhancement Courses

1. Skill Enhancement Courses have a significant theoretical component therefore theory workload is assigned to the course but the teaching of these courses should focus on practical application, with the goal of developing practical skills and knowledge as the final outcome.
2. There shall be no theory examination for Skill Enhancement Courses (SEC-1, SEC-2).
3. The evaluation of Skill Enhancement Courses should be entirely based on college internal assessment, meaning that the assessment will be carried out by the college's respective course incharge, rather than by an external entity.
4. To assess the students' understanding and skills in Skill Enhancement Courses, they should demonstrate their acquired skill through hands-on experience, practical work, projects, and case studies. There should be one assessment for each unit and an additional assessment at the end of the semester.

5. Records of each assessment should be maintained by the college's respective course incharge and should be readily made available upon request.
6. At the end of the semester, the consolidated marks should be submitted to the University for Inclusion in the student's mark sheet, which will contribute towards their final grade.

7. The university should generate the mark list for Skill Enhancement Courses, similar to the internal assessment mark list. The mark list should be downloaded, filled with the consolidated marks of all assessments, and submit along with the internal marks list.

Pattern of Question Paper (Theory)

B. Sc. (Computer Science) Semester - III and IV

Course Code -----

Paper Number -----

Title of Paper -----

Time : 1.30 Hrs.

Max Marks: 40

N.B.

1. Attempt All Questions.
2. All questions carry equal marks.
3. Illustrate your answer with suitable labelled diagram.

Q.1. Multiple choice questions / Fill In the Blanks / Terms / Definition / One Line

Answer questions.

(10 Marks)

- 1)
- 2)
- 3)
- 4)
- 5)
- 6)
- 7)
- 8)
- 9)
- 10)

Q.2. Long answer question.

(10 Marks)

OR

Long Answer question

Q.3. Long answer question

(10 Marks)

OR

Short answer questions

- a)
- b)

Q.4. Short Notes on any TWO of the following:-

(10 Marks)

- a)
- b)
- c)
- d)

Pattern of Question Paper(Practical)

B. Sc. (Computer Science) Semester - III and IV

Course Code -----

Paper Number -----

Title Of Paper -----

Time: 3:00 Hrs.

Max Marks : 100 (UA:80+IA:20)

N.B.

1. Attempt All Questions.
2. All questions carry equal marks.
3. Illustrate your answer with suitable labelled diagram

Section A

Q:1 Experiment based on CS-313 P (25 Marks)

- a) Question / Experiment– 35 Marks
- b) Viva / Oral – 05 Marks
- c) Internal Evaluation : 07 Marks
- d) Record book : 03 Marks

Section B

Q:2 Experiment based on CS-413 P (50 Marks)

- e) Question / Experiment– 35 Marks
- f) Viva / Oral – 05 Marks
- g) Internal Evaluation : 07 Marks
- h) Record book : 03 Marks

Dr. Babasaheb Ambedkar Marathwada University, Aurangabad

Choice Based Credit System (CBCS) Curriculum

For

Faculty of Science and Technology

Course Structure (Second Year)

B. Sc. (Computer Science) Three Year Under Graduate Degree Program

Second Year

Course Type	Semester – III		Course Type	Semester - IV	
	Name of Paper	Credit		Name of Paper	Credit
Core Course VII (DSC-I C) Core Course (CC) 7 Credits	Object Oriented Programming Using C++	2	Core Course X (DSC-I D) Core Course (CC) 7 Credits	Core Java	2
	Relational Database Management System	2		Computer Graphics	2
	Lab Course	1.5		Lab Course	1.5
	Lab Course	1.5		Lab Course	1.5
Core Course VIII (DSC-II C) Core Course (CC) 7 Credits	Linux Operating System	2	Core Course XI (DSC-II D) Core Course (CC) 7 Credits	Basics of Android OS	2
	Advanced Data Structure	2		Computer Networks	2
	Lab Course	1.5		Lab Course	1.5
	Lab Course	1.5		Lab Course	1.5
Core Course IX (DSC-III C) Core Course (CC) 7 Credits	Computational Statistics Using R	2	Core Course XII (DSC-III D) Core Course (CC) 7 Credits	Data Analytics	2
	Web Fundamental	2		Open Source Web Application Development	2
	Lab Course	1.5		Lab Course	1.5
	Lab Course	1.5		Lab Course	1.5
Skill Enhancement Course (SEC-1) 01 Course, 2 credit each	SEC-1 (Any one of the skill to be chosen out of two) (A) - Office Automation (B) - Critical Thinking	2	Skill Enhancement Course (SEC-2) 01 Course, 2 credit each	SEC-2 (Any one of the skill to be chosen out of two) (C)- Basic Python Programming (D)- Emotional Intelligence	2
Ability Enhancement Compulsory Courses (AECC), 02 Course, 3 credit each	Communication Skill in English-III	3	Ability Enhancement Compulsory Courses (AECC), 02 Course 3, credit each)	Communication Skill in English-IV	3
	Marathi/Hindi/Sanskrit/Urdu/Arabic - (SL-III) A student can opt for one of these languages	3		Marathi/Hindi/Urdu/Arabic (SL-IV) A Student can opt for of these languages	3
Non-Credit Course	-----	-----	Non-Credit Course	Environment Studies	--
Total Credit	29			29	

Dr. Babasaheb Ambedkar Marathwada University, Aurangabad

Choice Based Credit System (CBCS) Curriculum

For

Faculty of Science and Technology

Course Structure and Scheme of Examination (Second Year)

B. Sc. (Computer Science) Three Year Under Graduate Degree Program

Semester-III

Course Type	Course Code	Course Title	Total Periods (Teaching Periods / Week)	Credits	Scheme of Examination			
					UA	IA	Max Marks	Min Marks
Core Course VII (DSC-I C) Core Course (CC)	CS-311T	Object Oriented Programming Using C++	45 (3/per week)	2	40	10	50	20
	CS-312T	Relational Database Management System	45 (3/per week)	2	40	10	50	20
	CS-313P	Lab Course (based on CS-311T)	45 (3/per week)	1.5	40	10	50	20
	CS-314P	Lab Course (based on CS-312T)	45 (3/per week)	1.5	40	10	50	20
Core Course VIII (DSC-II C) Core Course (CC)	CS-321T	Linux Operating System	45 (3/per week)	2	40	10	50	20
	CS-322T	Advanced Data Structure	45 (3/per week)	2	40	10	50	20
	CS-323P	Lab Course (based on CS-321T)	45 (3/per week)	1.5	40	10	50	20
	CS-324P	Lab Course (CS-322T)	45 (3/per week)	1.5	40	10	50	20
Core Course IX (DSC-III C) Core Course (CC)	CS-331T	Computational Statistics Using R	45 (3/per week)	2	40	10	50	20
	CS-332T	Web Fundamental	45 (3/per week)	2	40	10	50	20
	CS-333P	Lab Course (based on CS-331T)	45 (3/per week)	1.5	40	10	50	20
	CS-334P	Lab Course (based on CS-332T)	45 (3/per week)	1.5	40	10	50	20
Skill Enhancement Course (SEC-1)**	CS-341	SEC-1 (Any one of the skill to be chosen out of two) (A) - Office Automation (B) – Critical Thinking	45 (3/per week)	2	-	50	50	20
Ability Enhancement Compulsory Courses (AECC-3)	CS-351T	Communication Skill in English-III	45 (3/per week)	3	40	10	50	20
	CS-361T	Marathi/Hindi/Sanskrit/Urdu/Arabic - (SL-III) A student can opt for one of these languages	45 (3/per week)	3	40	10	50	20
Non Credit Course								
45 Period Per week				29	560	190	750	300

*DCS – discipline Specific core courses

Total Credit for Semester III : 29 (Theory : 20 : Laboratory : 9)

****Refer Important note on Page 2 Related to Skill Enhancement Course Assessment**

Dr. Babasaheb Ambedkar Marathwada University, Aurangabad

Choice Based Credit System (CBCS) Curriculum

For

Faculty of Science and Technology

Course Structure and Scheme of Examination (Second Year)

B. Sc. (Computer Science) Three Year Under Graduate Degree Program

Semester-IV

Course Type	Course Code	Course Title	Total Periods (Teaching Periods/ Week)	Credits	Scheme of Examination			
					UA	IA	Max Marks	Min Marks
Core Course X (DSC-I D) Core Course (CC)	CS-411T	Core Java	45 (3/per week)	2	40	10	50	20
	CS-412T	Computer Graphics	45 (3/per week)	2	40	10	50	20
	CS-413P	Lab Course (based on CS-411T)	45 (3/per week)	1.5	40	10	50	20
	CS-414P	Lab Course (based on CS-412T)	45 (3/per week)	1.5	40	10	50	20
Core Course XI (DSC-II D) Core Course (CC)	CS-421T	Basics of Android OS	45 (3/per week)	2	40	10	50	20
	CS-422T	Computer Networks	45 (3/per week)	2	40	10	50	20
	CS-423P	Lab Course (based on CS-421T)	45 (3/per week)	1.5	40	10	50	20
	CS-424P	Lab Course (CS-422T)	45 (3/per week)	1.5	40	10	50	20
Core Course XII (DSC-III D) Core Course (CC)	CS-431T	Data Analytics	45 (3/per week)	2	40	10	50	20
	CS-432T	Open Source Web Application Development	45 (3/per week)	2	40	10	50	20
	CS-433P	Lab Course (based on CS-431T)	45 (3/per week)	1.5	40	10	50	20
	CS-434P	Lab Course (based on CS-432T)	45 (3/per week)	1.5	40	10	50	20
Skill Enhancement Course (SEC-2)**	CS-441	SEC-2 (Any one of the skill to be chosen out of two) (C) - Basic Python Programming (D) – Emotional Intelligence	45 (3/per week)	2	-	50	50	20
Ability Enhancement Compulsory Courses (AECC-4)	CS-451T	Communication Skill in English-IV	45 (3/per week)	3	40	10	50	20
	CS-461T	Marathi/Hindi/Sanskrit/Urdu/A rabic - (SL-IV) A student can opt for one of these languages	45 (3/per week)	3	40	10	50	20
Non Credit Course	CS-471T	Environment Studies	45 (3/per week)					
48 Period Per week				29	560	190	750	300

*DCS – discipline Specific core courses

Total Credit for Semester IV : 29 (Theory : 20 : Laboratory : 9)

****Refer Important note on Page 2 Related to Skill Enhancement Course Assessment**



B. Sc. (Computer Science)

Semester - III

Curriculum for semester III

Course Code: CS-311T	Course Title:- Object Oriented Programming Using C++
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: 1. Basic Programming Language Construct (Like looping and decision making) using C 2. Functions and Structures in C	
Learning Objectives 1. Understand principles of OOP Using C++. 2. Design and implement OOP programs using C++. 3. Understand classes, objects, inheritance, and polymorphism using C++. 4. Develop Object Oriented Programming skills and gain practical experience.	
Learning Outcomes After Completion of the Course students will be able to 1. Apply OOP principles to design efficient and scalable programs.. 2. Use C++ features to create complex and extensible programs. 3. Develop generic and reusable code using Polymorphism.	
Unit -I: Introduction to C++ (10 Periods) Input- output in C++, Data Types C++, and derive data types. The void data type, Type Modifiers, Typcasting, and Constant in C++, Operators in C++, Precedence of Operators, and Strings.	
Unit -II: Structures and Functions in C++ (10 Periods) Parts of Function, User- defined Functions, Value- Returning Functions, void Functions, Value Parameters, Function overloading, Virtual Functions. Structure in C++.	
Unit -III: Introduction to Oops using C++ (10 Periods) Object Oriented Technology, Advantages of OOP. Class, Build- in Operations on Classes, Assignment Operator and Classes, Class Scope, Reference parameters and Class Objects (Variables), Member functions, Accessor and Mutator Functions, Constructors, default Constructor, Destructors.	
Unit -IV: Overloading, Templates and Inheritance (10 Periods) Operator Overloading, Function Overloading, Function Templates, Class Templates. Single and Multiple Inheritance, virtual Base class, Abstract Class, Pointer and Inheritance, Overloading Member Function. Friend Function.	
Unit-V: Test and Tutorials (05 Periods) In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).	

Textbook

1. Object Oriented Programming with C++, 3/e by E. Balaguruswamy, Tata McGraw Hill.
2. Starting Out with Object Oriented Programming in C++, by Tony Gaddis, Wiley India.

References:

1. Mastering C++, 1/e by Venugopal, Tata McGraw Hill.
2. The C++ Programming language 3/e by Bjarne Stroustrup, Pearson Education.
3. C++, How to Programme, 4e, by Deitel, Pearson Education.
4. Big C++ by Cay Horstmann, Wiley India.

E-Resources

1. Cplusplus.com: A comprehensive online resource for learning C++ programming, including tutorials, code examples, and a reference guide.
<https://cplusplus.com/>
2. Codecademy: An online learning platform that offers an interactive C++ course that covers OOP concepts.
<https://www.codecademy.com/resources/docs/cpp>

Course Code: CS-312T	Course Title:- Relational Database Management System
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: Database Management System Concepts from Course CS-216T	
Learning Objectives <ol style="list-style-type: none">1. Understand the basic concepts of Relational Database Management System (RDBMS).2. Learn to design and create a relational database schema using SQL.3. Explore the functionalities of RDBMS and learn to implement them for data manipulation and retrieval.4. Understand the concepts of normalization and apply them to eliminate data redundancy and improve data integrity.	
Learning Outcomes After Completion of the Course students will be able to <ol style="list-style-type: none">1. Design and create a relational database schema using SQL.2. Implement various RDBMS functionalities such as data insertion, deletion, modification, and retrieval.3. Demonstrate the ability to use SQL to write complex queries for data analysis and reporting.4. Understand the principles of database normalization and apply them to ensure data integrity and optimize database performance.	
Unit -I: (10 Periods) Relational Model: CODD's Rule- Relational Data Model - Key - Integrity – Relational Algebra Operations	

– Advantages and limitations – Relational Calculus – Domain Relational Calculus -QBE.

Unit -II: (10 Periods)

Structure of Relational Database. Introduction to Relational Database Design - Objectives – Tools – Redundancy and Data Anomaly – Functional Dependency - Normalization – 1NF – 2NF – 3NF –BCNF. Transaction Processing – Database Security.

Unit -III: (10 Periods)

SQL: Commands – Data types – DDL - Selection, Projection, Join and Set Operations – Aggregate Functions – DML – Modification - Truncation - Constraints – Subquery.

Unit -IV: (10 Periods)

PL/SQL: Structure - Elements – Operators Precedence – Control Structure – Iterative Control -Cursors - Procedure - Function - Packages – Exceptional Handling - Triggers.

Unit-V: Test and Tutorials (05 Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. S. Sumathi, S. Esakkirajan, “Fundamentals of Relational Database Management System”, Springer International Edition 2007.

References:

1. Abraham Silberchatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, McGrawHill 2019, 7th Edition.
2. Alexis Leon & Mathews Leon, “Fundamentals of DBMS”, Vijay Nicole Publications 2014, 2nd Edition.

E-Resources

1. SQLBolt (<https://sqlbolt.com/>): This is a free interactive tutorial that teaches SQL commands using simple exercises and examples.
2. W3Schools SQL (<https://www.w3schools.com/sql/>): This is a comprehensive and free online resource for learning SQL and database management.

Course Code: CS-313P	Course Title: Lab Course (based on CS-311T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-311T**

(The teacher can add three practical examples based on each unit as per their choice and feasibility)

1. Write a C++ program to implement basic data types and operators.
2. Write a C++ program to implement control structures like loops and conditional statements.
3. Write a C++ program to implement a calculator using basic arithmetic operators and control structures.
4. Write a C++ program to implement functions and function overloading.
5. Write a C++ program to demonstrate implementation of structures in C++.
6. Write a C++ program to implement a function that converts a string to uppercase using string functions.
7. Write a C++ program to implement a function that converts a string to uppercase using pointers and arrays.
8. Write a C++ program to implement a class that represents a number with functions for addition, subtraction, multiplication, and division. (Operator overloading)
9. Write a C++ program to implement a class that represents a book with functions for adding, deleting books in a library system. (Introduction to classes and objects)
10. Write a C++ program to implement a class that represents a bank account with functions for deposit, withdraw, and balance check. (Basics of class and object creation)
11. Write a C++ program to implement a class that represents a date with functions for setting and getting the date and calculating the difference between two dates. (Function overloading)
12. Write a C++ program to implement a class hierarchy that includes a base class called "Vehicle" and two derived classes called "Car" and "Motorcycle" with functions for displaying their respective features. (Inheritance and polymorphism)
13. Write a C++ program to implement a class hierarchy that includes a base class called "Shape" and two derived classes called "Circle" and "Rectangle" with functions for calculating their respective areas and perimeters. (Inheritance and polymorphism)
14. Write a C++ program to implement a class that represents a date with functions for setting and getting the date and calculating the difference between two dates. (Function overloading)
15. Write a C++ program to implement the concept of friend function.

Course Code: CS-314P	Course Title: Lab Course (based on CS-312T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-312T**

(The teacher can make use of MySQL or Oracle for laboratory practice and add three practical examples based on each unit as per their choice and feasibility)

1. Create a database and tables using SQL commands
2. Insert data into tables using SQL queries
3. Update existing data in tables using SQL queries
4. Delete data from tables using SQL queries
5. Use SELECT statement to retrieve data from tables
6. Use WHERE clause to filter data in SELECT statements
7. Use GROUP BY and HAVING clauses to aggregate data in SELECT statements
8. Join multiple tables using INNER JOIN and OUTER JOIN
9. Use subqueries to retrieve data from multiple tables
10. Create views to simplify complex SQL queries
11. Create indexes to improve query performance
12. Use data normalization techniques to design and create efficient database schemas
13. Implement foreign keys and referential integrity constraints in database schemas

14. Use transactions to ensure data consistency and atomicity in database operations
15. Backup and restore databases using SQL commands and tools

Course Code: CS-321T	Course Title:- Linux Operating Systems
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: Operating System I & II Courses (CS-113T & CS-213T)	
Learning Objectives <ol style="list-style-type: none"> 1. To learn basics of Linux Operating System, its components, features and flavors 2. To learn basic and common Linux commands 3. To learn to set ownership and permissions of the files and directories 4. To learn to manipulate files/directories. 5. To learn working in Vi Editor 	
Learning Outcomes After Completion of the Course students will be able to <ol style="list-style-type: none"> 1. Understand the various features and distributions of Linux OS. 2. Ability to execute basic Linux commands. 3. Ability to set ownership and permissions for files/directories. 4. Ability to use the Vi Editor. 	
Unit -I: History and Development of Linux (10 Periods) A Brief History of Linux, Basic features of Linux OS, components of Linux System, Benefits of Linux, Acquiring and Using Linux, Examining Linux Distributions, Installation notes, Linux Loader, Linux kernel, Linux file system.	
Unit -II: System Access & User Accounts (10 Periods) System Access and User Accounts -Logging In and out Using the Linux System, Creating Additional User Accounts, Creating & Managing Groups, and Managing Users Linux Commands.	
Unit -III: File System & File Permissions (10 Periods) Introduction to The File System and Working with Linux Permissions , File System Navigation, Managing The File System Understanding Permissions, Changing File And Directory Permissions, Changing Default Permissions And Ownership	
Unit -IV: Using Editors (10 Periods) Using The Vi Editor, Studying Other Editors, Redirection, and Introduction to Programming In C Using Linux (gcc).	
Unit-V: Test and Tutorials (05 Periods) In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).	

Textbook

1. Linux for Beginners: The Ultimate Guide To The Linux Operating System & Linux Commands 1st Edition By Adam Vardy.
2. LINUX: The Ultimate Step by Step Guide to Quickly and Easily Learning Linux by TED DAWSON

References:

1. McAllister, S, Use Linus-10, Pearson Education, 2006 ISBN-81-7808-488-0 PHI.
2. Ball, Using Linux, PHI, 1998. ISBN-10: 0789716232
3. Das, UNIX: Concepts and Applications (4th Ed), TMH, 2006 ISBN 13: 9780070635463.
4. Foster Johnson, Welch, Anderson, Beginning Shell Scripting, Wiley India (Wrox), 2006 ISBN- 10: 0764583204
5. Neil Mathew, Richard Stones, Beginning Linux Programming (3rd Ed), Wiley India (Wrox), 2006 ISBN: 978-0-470-14762-7
6. Peterson, Linux: Complete Reference (5th Ed), Peterson, TMH. ISBN 10: 0070222940

E-Resources

1. Linux Journey - <https://linuxjourney.com/> - It is a free interactive online tutorial that covers all the basics of Linux with a series of short lessons.
2. edX Linux Course - <https://www.edx.org/learn/linux> - edX offers a free online course on Linux that covers the fundamentals of Linux, the command-line interface, and basic scripting.
3. Linux Tutorial - https://www.tutorialspoint.com/unix_commands/index.htm - This is a comprehensive tutorial that covers all the basic concepts of Linux, including command-line interface, file management, and shell scripting.

Course Code: CS-322T	Course Title:- Advance Data Structure
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: Data Structure Courses (CS-211T)	
Learning Objectives <ol style="list-style-type: none">1. To provide knowledge linked list, its types and its in computer memory.2. To familiarize with non-linear data structures.3. To provide knowledge on how advance data structures are implemented and processed.4. To equip with the implementation techniques of complex algorithms of insertion, deletion and modification of data stored in advance data structures.5. To provide knowledge of the functioning of dynamic data structures like heaps binary search trees.	
Learning Outcomes After Completion of the Course students will be able to <ol style="list-style-type: none">1. Understand linked-lists and non-linear data structures like trees and graphs.	

2. Program Linked list and non-linear data structure's implementation in memory.
3. Select appropriate data structures and algorithms for problems and to justify their choice.
4. Understand advance algorithms like heaps, Kruskal's Algorithm, and Prim's Algorithm.

Unit -I: Linked List (10 Periods)

Drawbacks of Arrays, Introduction to Linked lists. Types of Linked Lists, Representation of Linked List in Memory, Operations on Singly Linked Lists (Traversing, Insertion, Deletion and modification), Doubly Linked List, Representation of Doubly Linked List in Memory, Operations on doubly Linked Lists (Traversing, Insertion, Deletion and modification).

Unit -II: Trees (10 Periods)

Introduction and key terminology, Binary Trees Binary Tree Creation and Traversal Using Arrays, Binary Tree Creation and Traversal Using Pointers, Expression Trees, traversing binary tree recursively and non-recursively (pre-order, in order, post order traversal). Application of trees (binary search tree).

Unit -III: Graphs(10 Periods)

Introduction and key terminology, graph representation in memory (static and dynamic), traversing a graph (breath first search, depth first search), spanning tree. Kruskal's Algorithm, Prim's Algorithm

Unit -IV: Advance Trees (10 Periods)

Heaps, Min/ Max Heap, Binomial Heap, Fibonacci Heap, Heap Sort, B Tree, B+ Tree.

Unit-V: Test and Tutorials (05Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. Data Structures using C, by Seema Threja, 2nd Edition, Oxford Press.
2. Lipschutz: Schaum's outline series Data structures Tata McGraw-Hill

References:

1. Fundamentals of Data Structures in C, by Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed
2. Design & Analysis of computer Algorithms by Alfred Aho, John Hopcroft and Jeffery Ullman
3. Introduction to Algorithms by Thomas Corman et.al

E-Resources

1. Coursera: Data Structures and Algorithms Specialization Link:
<https://www.coursera.org/specializations/data-structures-algorithms> This is a series of courses offered by the University of California San Diego on Coursera. It covers topics like algorithmic analysis, graph algorithms, data structures and dynamic programming.
2. Data Structures and Algorithms in C++ by Adam Drozdek Link:
<https://www.pdfdrive.com/data-structures-and-algorithms-in-c-e16544168.html> This is a free ebook that covers data structures and algorithms using C++. It includes topics like arrays, linked lists, stacks, queues, trees, sorting and searching algorithms, and graph algorithms.

Course Code: CS-323P	Course Title: Lab Course (based on CS-321T)
-----------------------------	--

Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-321T**

(The teacher can make use of any flavour of Linux distribution and add few more practical based on each unit)

1. Access: Logging In. Linux Commands. Getting Help. Obtaining Information about Your System.
2. Starting and Stopping Linux: Shutting Down a Linux System, Booting a Linux System.
3. Demonstration of Linux commands with attributes: - pwd, cd, ls, more, less, echo, clear, kill, ps, man, cal, date, who, who am I, WC, mkdir, rmdir, rm, sort.
4. File and File Permission: Creation of Files, and changing their permission (Cat,vi, Chmod)
5. Archiving Files: Archiving Files with tar
6. Write a shell script to display first 20 terms of Fibonacci series.
7. Write a shell script to display current time of system and display the message according to the time.
8. Write a shell script to check the user is login or not and say hello.
9. Write a shell script to calculate factorial of a number
10. Using filters & redirections: create new processed files (Using Head, tail, cut, paste etc. create resultsheet/salariesheet)
11. Develop a C Program In Linux to find out 20 terms of Fibonacci series.
12. Develop a C Program In Linux to calculate factorial of a number

Course Code: CS-324P	Course Title: Lab Course (based on CS-322T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-322T**

(The teacher can make use of any language to implement these programs but are suggested to use either C or C++. Also teacher can add few more practical based on each unit)

Practical No	Details
	Implement Singly Linked List
1	Write a program to create a singly linked, add few nodes, and display the same.
2	Write a program to create a singly linked, add new node at the beginning of the linked list, and display list before and after adding new node.
3	Write a program to create a singly linked, add new node at the end of the linked list, and display list before and after adding new node.
4	Write a program to create a singly linked, delete node at the beginning of the linked list, and display list before and after deletion.
5	Write a program to create a singly linked, delete the last node of the linked list, and display list before and after deletion.

6	Write a program to create a singly linked, add few nodes, modify node at a specific location, and display the list before and after modification.
	Implement Doubly Linked List
7	Write a program to create a doubly linked, add few nodes, and display the same.
8	Write a program to create a doubly linked, add new node at the beginning of the linked list, and display list before and after adding new node.
9	Write a program to create a doubly linked, add new node at the end of the linked list, and display list before and after adding new node.
10	Write a program to create a doubly linked, delete node at the beginning of the linked list, and display list before and after deletion.
11	Write a program to create a doubly linked, delete the last node of the linked list, and display list before and after deletion.
12	Write a program to create a doubly linked, add few nodes, modify node at a specific location, and display the list before and after modification.
	Implement Trees
13	Write a program to create a binary tree of degree 3, display each node.
14	Write a program to create a binary tree of degree 3, and search an element in the tree.
	Implement Graphs:
15	Write a program to implement the concept of breath first search.
16	Write a program to implement the concept of depth first search.
	Implement Advance Trees:
17	Write a program to create a heap tree
18	Write a program to demonstrate the Prim's algorithm

Course Code: CS-331T	Course Title:- Computational Statistics Using R
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: Programming language basics.	
Learning Objectives <ol style="list-style-type: none"> 1. To introduce students to the fundamentals of statistics and their applications in various fields. 2. To develop proficiency in using the R programming language for data analysis and visualization. 3. To teach students essential statistical techniques, including descriptive statistics, inferential statistics, and regression analysis. 4. To enable students to apply statistical methods to real-world datasets and interpret the results. 	
Learning Outcomes By the end of the course, students will be able to: <ol style="list-style-type: none"> 1. Understand the core concepts and methods in statistics, and recognize their importance in various 	

disciplines.

2. Effectively use the R programming language to manage, analyze, and visualize data.
3. Apply appropriate statistical techniques, such as hypothesis testing and regression analysis, to answer research questions and make data-driven decisions.
4. Analyze real-world datasets using statistical methods and R, interpret the results, and communicate their findings to both technical and non-technical audiences.

Unit -I: Introduction to Statistics and R Language (10 Period)

1. **Importance of Statistics in Various Fields (01 Period):** Definition and purpose of statistics, Applications of statistics in different fields, such as: Business and economics (e.g., market research, financial analysis), Healthcare (e.g., clinical trials, epidemiology), Social sciences (e.g., psychology, sociology, political science), Natural sciences (e.g., physics, chemistry, biology), Engineering (e.g., quality control, reliability engineering).
2. **Introduction to R Programming Language and R-Studio (05 Period): Overview of R and its advantages:** Installing R and R-Studio, Navigating the R-Studio interface (console, script editor, environment, plots, and help), R packages and CRAN repository, Basic R Syntax, Data Types, and Operations, R syntax and expressions, Data types: numeric, character, logical, factor, and date/time, **Data structures:** vector, matrix, list, and data frame, Basic R operations: arithmetic, relational, and logical, **Control structures:** if-else, for loops, and while loops, Functions: built-in and user-defined.
3. **Measures of Central Tendency (Mean, Median, Mode) (02 Period):** Definition and properties of mean, median, and mode. Calculation of mean, median, and mode using R functions: mean, median, and mode, **Measures of Dispersion (Range, Variance, Standard Deviation):** Definition and properties of range, variance, and standard deviation, Calculation of range, variance, and standard deviation using R functions: range, var, and sd.
4. **Introduction to Data Visualization (02 Period):** Importance of data visualization, Types of data visualizations (e.g., bar chart, pie chart, line chart, scatter plot, histogram, box plot), Basic principles of good data visualization.

Unit -II: Probability and Data Distributions (10 Periods)

1. **Basics of Probability Theory:** Definition of probability and its properties, Sample space, events, and outcomes, Basic rules of probability: addition rule, multiplication rule, and conditional probability, Independent and dependent events, Bayes' theorem
2. **Discrete Probability Distributions:** Introduction to discrete probability distributions, Probability mass function (PMF), Expected value and variance of discrete random variables, Binomial distribution: definition, properties, and applications, R functions: dbinom, pbinom, qbinom, rbinom, Poisson distribution: definition, properties, and applications, R functions: dpois, ppois, qpois, rpois.
3. **Continuous Probability Distributions:** Introduction to continuous probability distributions, Probability density function (PDF) and cumulative distribution function (CDF), Expected value and variance of continuous random variables, Normal distribution: definition, properties, and applications, R functions: dnorm, pnorm, qnorm, rnorm, Exponential distribution: definition, properties, and applications, R functions: dexp, pexp, qexp, rexp
4. **Working with Probability Distributions in R:** Generating random samples from discrete and continuous distributions, Estimating distribution parameters from data, Computing probabilities and percentiles using R functions, Visualizing probability distributions: histograms, density plots, and empirical CDFs, Fitting probability distributions to data using R packages like fitdistrplus

Unit -III: Basic Inferential Statistics: (10 Periods)

1. **Sampling and Sampling Distributions:** Definition and importance of sampling, Types of sampling methods (e.g., simple random sampling, stratified sampling, cluster sampling), Sampling distribution and its properties, Central Limit Theorem and its implications, Standard error of the mean and its calculation using R: sd and length functions.
2. **Confidence Intervals:** Definition and purpose of confidence intervals, Interpretation of confidence intervals. Calculation of confidence intervals for population mean (using t-distribution), R functions: t.test, qt, and manual calculation, Calculation of confidence intervals for population proportion, R functions: prop.test and manual calculation.
3. **Hypothesis Testing:** t-test and chi-square test: Definition and purpose of hypothesis testing, Null hypothesis and alternative hypothesis, Type I and Type II errors, significance level, and power, One-sample t-test, two-sample t-test, and paired t-test, R functions: t.test, Chi-square test for goodness-of-fit and independence, R functions: chisq.test
4. **Introduction to Linear Regression:** Definition and purpose of linear regression, Simple linear regression model: assumptions and parameters, Estimation of parameters using the least-squares method, Interpretation of the regression coefficients and the coefficient of determination (R-squared), R functions for linear regression: lm, summary, confint, predict, and plot

Unit -IV: Data Analysis and Visualization using R (10 Periods)

1. **Data Visualization Techniques in R: Histograms:** visualizing the distribution of a continuous variable, R functions: hist, **Box plots:** displaying the five-number summary of a continuous variable, R functions: boxplot, **Scatter plots:** visualizing the relationship between two continuous variables, R functions: plot, **Bar charts:** representing the frequency or proportion of categorical variables, R functions: barplot, table
2. **Analysing Real-World Datasets and Case Studies**
 - Choosing appropriate datasets for practice and analysis (e.g., from sources like Kaggle, UCI Machine Learning Repository, or government websites)
 - Steps for analysing real-world datasets:
 1. **Data exploration and pre-processing:** handling missing values, outliers, and data transformations
 2. **Descriptive statistics:** calculating measures of central tendency, dispersion, and visualizing the data
 3. **Inferential statistics:** applying hypothesis testing and regression analysis to answer research questions
 4. **Interpretation and communication of results**

Encourage students to work on real-world case studies related to their interests or field of study

Unit-V: Test and Tutorials (05 Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. "Introductory Statistics with R" by Peter Dalgaard (Springer, 2nd Edition, 2008)
2. "Discovering Statistics Using R" by Andy Field, Jeremy Miles, and Zoë Field (SAGE Publications, 2012)

References:

1. "Statistics for Business and Economics" by Paul Newbold, William Carlson, and Betty Thorne.
2. "Probability and Statistics for Engineers and Scientists" by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, and Keying Ye.
3. "Introduction to Probability and Statistics" by William Mendenhall, Robert J. Beaver, and Barbara M. Beaver.

E-Resources

1. "R Programming for Data Science" by Roger D. Peng
 - Download: <https://bookdown.org/rdpeng/rprogdatascience/>
 - This book focuses on R programming, providing a solid foundation for students interested in learning R for data science and statistical analysis.
2. "An Introduction to Statistical Learning with Applications in R" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani
 - Download: <https://www.statlearning.com/>
 - This book covers various statistical learning methods, including linear regression, classification, and clustering, with a focus on applications in R.
3. "The Art of R Programming" by Norman Matloff
 - Download: <https://www.nostarch.com/artofr.htm> (Sample PDF available)
 - This book covers the fundamentals of R programming, data structures, and functions. While not strictly focused on statistics, it provides a strong foundation in R programming for statistical analysis.
4. "R for Data Science" by Hadley Wickham and Garrett Grolemund
 - Download: <https://r4ds.had.co.nz/> (PDF available via the link "Get the book" on the top-right corner)

Course Code: CS-332T	Course Title:- Web Fundamentals
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: There are no prerequisites for this course	
Learning Objectives <ol style="list-style-type: none">1. Understand the basic concepts and principles of web technologies, including HTML, CSS, and JavaScript.2. Gain practical skills in creating responsive and accessible web designs.3. Learn how to validate web pages and follow web standards set by the W3C.4. Develop proficiency in manipulating the Document Object Model (DOM) using JavaScript.	
Learning Outcomes After Completion of the Course students will be able to <ol style="list-style-type: none">1. Develop functional and visually appealing web pages using HTML and CSS.2. Design responsive web layouts that adapt to different devices and screen sizes.	

3. Implement client-side scripts using JavaScript to add interactivity and dynamic behavior to web pages.
4. Validate and optimize web pages for accessibility, performance, and compliance with web standards.

Unit -I: Introduction (10 Periods)

1. Web Browsers: Introduction to web browsers, types of web browsers, how they work.
2. Web Servers: Introduction to web servers, types of web servers, how they work.
3. Client-side vs Server-side: Understanding the difference between client-side and server-side processing.
4. Web Standards: Introduction to web standards, W3C, HTML validation, CSS validation.
5. **HTML Syntax and Basic Tags:** study the basic syntax of an HTML document, including the doctype declaration, opening and closing tags, and nesting of elements. We will also explore fundamental HTML tags like <head>, <body>, <h1>-<h6>, <p>, <a>, and .
6. **Structure of an HTML Document:** delve into the standard structure of an HTML document, including the <!DOCTYPE> declaration, the <html> element, and the <head> and <body> sections. how to use comments and the proper organization of elements within the document.
7. **HTML Elements and Attributes:** learn about the different types of HTML elements, including block-level and inline elements. cover how to use attributes to provide additional information about an element, such as the 'src' attribute for images or the 'href' attribute for links.
8. **Semantic HTML:** explore the importance of using semantic elements in HTML5, such as <article>, <section>, <header>, <nav>, and <footer>, and how these elements can enhance the accessibility and search engine optimization of web pages.
9. **HTML5:** study the new features and improvements introduced in HTML5, including multimedia elements like <video> and <audio>, new form input types and attributes, and JavaScript APIs for advanced functionality.
10. **Lists:** learn how to create ordered and unordered lists using the and elements,
11. **Links and Navigation:** study how to create different types of links using the <a> element, including internal, external, and anchor links, as well as email and telephone links.

Unit -II: CSS Fundamentals (10 Periods)

1. **Introduction to CSS**
 Understanding the purpose of CSS
 Syntax and structure of CSS rules
 Applying CSS: inline, internal, and external stylesheets
 Linking a CSS file to an HTML document using the <link> element
2. **Basic Selectors**
 Element, class, and ID selectors
 Universal and attribute selectors
 Grouping and chaining selectors
 Understanding selector specificity
3. **Advanced Selectors and Combinators**
 Descendant, child, and sibling combinators
 Pseudo-classes: **:hover**, **:active**, **:visited**, **:first-child**, **:last-child**, and **:nth-child**
 Pseudo-elements: **::before**, **::after**, and **::first-letter**
 Attribute selectors with various matching patterns
4. **Box Model: Basics**
 Understanding the CSS box model (content, padding, border, margin)
 Setting width and height of elements
 Managing overflow and scrollbars
5. **Box Model: Padding, Margin, and Border**
 Setting padding, margin, and border properties
 Using shorthand notation for padding, margin, and border

box-sizing property and its values (**content-box**, **border-box**)

6. **Layout and Positioning: Display Property**

Understanding the **display** property (block, inline, inline-block)

Using the **display** property to create layouts

Controlling element visibility with **display: none** and **visibility: hidden**

7. **Layout and Positioning: Floats and Positioning**

Creating multi-column layouts with **float**

Clearing floats with the **clear** property

Static, relative, absolute, and fixed positioning

8. **Layout and Positioning: Flexbox**

Introduction to the CSS Flexbox layout system

Defining a flex container and flex items

Controlling the direction, alignment, and order of flex items

Handling flexible sizes and growing/shrinking of items

9. **Layout and Positioning: CSS Grid**

Introduction to the CSS Grid layout system

Defining a grid container and grid items

Setting up grid columns, rows, and gaps

Positioning grid items and controlling their size

10. **Review and Best Practices**

Review of key concepts covered in the course

Organizing and structuring CSS code

CSS naming conventions and methodologies (e.g., BEM)

Tips for writing maintainable and efficient CSS

Unit -III: Advanced HTML and CSS Techniques (10 Periods)

1. Advanced HTML: Tables, forms, multimedia, accessibility, SEO.
2. Responsive Web Design: Understanding responsive design principles, media queries, fluid grids, responsive images.
3. CSS3: Advanced CSS3 techniques, transitions, animations, transforms, and gradients.

Unit -IV: JavaScript Fundamentals (10 Periods)

1. Introduction to JavaScript: Basic concepts, syntax, and usage.
2. Control Structures and Functions: Variables, data types, operators, control structures, functions, and arrays.
3. DOM Manipulation: Accessing and manipulating the Document Object Model (DOM) using JavaScript.
4. Events and Event Handling: Handling user events, event propagation, and delegation.

Unit-V: Test and Tutorials (05 Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. WEB TECHNOLOGIES 2010 by Uttam K.
2. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" 5th Edition by Jennifer Niederst Robbins
3. "Responsive Web Design with HTML5 and CSS" by Ben Frain
<https://vdoc.pub/download/responsive-web-design-with-html5-and-css3-5vk0jcsnmdv0>

References:

1. "HTML and CSS: Design and Build Websites" by Jon Duckett
2. "Web Design with HTML, CSS, JavaScript and jQuery Set" by Jon Duckett
3. "Web Development and Design Foundations with HTML5" by Terry Felke-Morris
4. "Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages" by Elisabeth Robson and Eric Freeman.

E-Resources

1. W3Schools (<https://www.w3schools.com/>) - Provides comprehensive tutorials and references for HTML, CSS, JavaScript, and other web technologies.
2. CSS Tricks: Responsive Design - <https://css-tricks.com/guides/responsive-design/>

Course Code: CS-333P	Course Title: Lab Course (based on CS-331T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-331T**

1. R Basics
How do you create and manipulate vectors, matrices, and data frames in R? Demonstrate using arithmetic and logical operators.
2. Importing Data
How do you import a dataset in CSV format into R? Show how to explore its structure, dimensions, and summary statistics.
3. Data Cleaning
How do you handle missing data, recode variables, and create new variables based on existing ones in R?
4. Descriptive Statistics
How do you calculate measures of central tendency and dispersion for a given dataset in R? What can you interpret from the results?
5. Data Visualization: Histograms
How do you create histograms for continuous variables in a dataset using R? What can you analyze from the shape of the distributions?
6. Data Visualization: Box Plots
How do you create box plots for continuous variables in a dataset using R? How can you compare distributions and identify outliers?
7. Data Visualization: Scatter Plots
How do you create scatter plots to visualize the relationship between two continuous variables in R? What can you explore about potential correlations?

8. **Data Visualization: Bar Charts**
How do you create bar charts to visualize the frequency or proportion of categorical variables in a dataset using R?
9. **Probability Distributions**
How do you generate random samples from binomial, Poisson, normal, and exponential distributions in R? How can you visualize the results using histograms?
10. **Confidence Intervals**
How do you calculate confidence intervals for population means and proportions using the t-distribution in R? How do you interpret the results?
11. **Hypothesis Testing: t-test**
How do you conduct one-sample, two-sample, and paired t-tests in R? How do you interpret the results and draw conclusions?
12. **Hypothesis Testing: Chi-square Test**
How do you conduct chi-square tests for goodness-of-fit and independence in R? How do you interpret the results and draw conclusions?
13. **Simple Linear Regression**
How do you fit a simple linear regression model to a dataset in R? How do you interpret the coefficients and assess the model's performance using R-squared?
14. **Model Diagnostics and Assumptions**
How do you check the assumptions of a linear regression model (normality of residuals, heteroskedasticity, multicollinearity) in R? What transformations or modifications can you perform if necessary?
15. **Multiple Linear Regression**
How do you fit a multiple linear regression model to a dataset in R? How do you interpret the coefficients and assess the model's performance using R-squared and adjusted R-squared?

Course Code: CS-334P	Course Title: Lab Course (based on CS-332T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-332T**

Unit I: Introduction

1. Setting up a local web server and creating a simple HTML webpage using a text editor.
2. Examining the HTTP request and response headers in the developer tools of a web browser.
3. Writing and testing a simple client-side JavaScript program using a web browser console.

4. Using W3C HTML validator to validate an HTML webpage.
5. Examining the difference between client-side and server-side processing using a simple form submission.

Unit II: CSS Fundamentals

1. Creating a simple webpage with basic CSS styling.
2. Experimenting with different CSS selectors and specificity to understand how they affect the styling of a webpage.
3. Creating a webpage with different layout and positioning techniques such as floats and flexbox.
4. Modifying the box model properties such as padding, margin, and border to achieve desired layouts.
5. Using CSS preprocessors like SASS to generate and compile CSS.

Unit III: Advanced HTML and CSS Techniques

1. Building a responsive website using fluid grids and media queries.
2. Creating a form with advanced HTML techniques like validation and accessibility features.
3. Implementing animations, transitions, and transformations using CSS3.
4. Experimenting with advanced CSS3 properties like gradients and filters.
5. Using accessibility tools to test and improve website accessibility.

Unit IV: JavaScript Fundamentals

1. Building a simple JavaScript application using control structures and functions.
2. Using JavaScript to manipulate the Document Object Model (DOM) and dynamically update the webpage.
3. Implementing event handling using JavaScript to create interactivity on a webpage.
4. Building a simple calculator application using JavaScript functions and event handling.
5. Using JavaScript libraries like jQuery to simplify and enhance DOM manipulation and event handling.

Course Code: CS-341T(A)	Course Title:- Office Automation
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: There are no prerequisites for this course	
Learning Objectives	
<ol style="list-style-type: none"> 1. To introduce the students to the concept of office automation and the benefits it provides. 2. To enable the students to use word processing tools for creating, formatting, revising, and sharing documents. 3. To enable the students to use spreadsheet and database management tools for data analysis and visualization. 4. To enable the students to use communication and collaboration tools while ensuring data security and privacy. 	

Learning Outcomes

After Completion of the Course students will be able to

1. Define and explain the scope and benefits of office automation.
2. Create and format documents using word processing tools.
3. Manage data using spreadsheet and database management tools.
4. Use communication and collaboration tools securely and maintain data privacy.

Unit -I: Introduction to Office Automation (10 Periods)

1. Definition and scope of office automation
2. Benefits of office automation
3. Overview of office automation tools and applications
4. History of office automation
5. Trends in office automation

Unit -II: Word Processing and Document Management (10 Periods)

1. Creating and formatting basic documents
2. Advanced formatting techniques (e.g. styles, templates, themes)
3. Working with tables and columns
4. Managing document content and structure
5. Reviewing and revising documents
6. Document sharing and collaboration
7. Automating document creation (e.g. mail merge, macros)

Unit -III: Spread sheet and Database Management (10 Periods)

1. Creating and managing basic spreadsheets
2. Advanced formatting techniques (e.g. conditional formatting, data validation)
3. Data analysis and visualization (e.g. charts, pivot tables)
4. Database management and design (e.g. creating tables, relationships, queries)
5. Importing and exporting data
6. Automating tasks (e.g. macros, scripts)

Unit -IV: Communication and Collaboration Tools; Security and Privacy in Office Automation (10 Periods)

1. Email and instant messaging basics
2. Advanced email features (e.g. filters, rules, signatures)
3. Online meetings and web conferencing basics
4. Advanced collaboration tools (e.g. shared calendars, task lists, project management)
5. Security threats and risks in office automation
6. Data protection and encryption basics
7. Best practices for secure communication and collaboration

Unit-V: Test and Tutorials (05 Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. "Office Automation: Principles and Practice" by Dr. R. K. Singla and Dr. N. P. Singh.
2. "Office Automation and Collaboration" by Prakash Rao

References:

1. "Office 2019 All-in-One For Dummies" by Peter Weverka (Wiley, 2018)
2. "Microsoft Office 2019 Inside Out" by Joe Habraken (Microsoft Press, 2019)
3. "Office 365 & Exchange Online: Essentials for Administration" by William Stanek (CreateSpace Independent Publishing Platform, 2017)
4. "Office 365 for Dummies" by Rosemarie Withee, Ken Withee, and Jennifer Reed (Wiley, 2019)
5. "The Ultimate Guide to Microsoft Office 365" by Sherri McLeish (Independently Published, 2021)

E-Resources

1. Office Automation - Overview - https://www.tutorialspoint.com/office_automation/office_automation_overview.htm
2. History and Development of Office Automation - <https://www.guru99.com/office-automation.html>
3. The Advantages of Office Automation - <https://smallbusiness.chron.com/advantages-office-automation-3077.html>
4. Microsoft Word Basics - <https://edu.gcfglobal.org/en/wordbasics/>
5. Advanced Microsoft Word - <https://edu.gcfglobal.org/en/advanced-word/>
6. Microsoft Excel Basics - <https://edu.gcfglobal.org/en/excelbasics/>
7. Advanced Microsoft Excel - <https://edu.gcfglobal.org/en/advanced-excel/>
8. Database Management Basics - <https://www.guru99.com/database-management-system.html>
9. Google Meet Basics - <https://edu.gcfglobal.org/en/google-meet/>
10. Microsoft Teams Basics - <https://edu.gcfglobal.org/en/microsoft-teams/>
11. Basic Internet Security - <https://www.gcflearnfree.org/internetsafety/basic-internet-security/>

Here are some practical exercises that align with your syllabus:

Unit -I: Introduction to Office Automation

1. Research different definitions of office automation and write a brief summary of your findings.
2. Discuss the scope of office automation in your own words and provide examples.
3. Make a list of benefits an office might experience from automation and explain each.
4. Use online resources to create a timeline detailing the history of office automation.
5. Research current trends in office automation and write a short report on three of them.
6. Identify an office process that could benefit from automation and explain how.
7. Create a presentation on a specific office automation tool of your choice.
8. Compare and contrast different office automation tools.
9. Prepare a case study of a company that has successfully implemented office automation.
10. Conduct a mock interview with a manager who has implemented office automation, focusing on the challenges and solutions they encountered.

Unit -II: Word Processing and Document Management

1. Create a basic document in a word processing application of your choice.
2. Use styles, templates, and themes to format a document.
3. Create a document that includes a table and formatted columns.
4. Use a word processing tool to rearrange and manage the content and structure of a document.
5. Use the review and revision features in your word processing tool to edit a document.
6. Demonstrate how to share a document and collaborate with others using an online platform.
7. Create a mail merge document.
8. Write and run a macro to automate a task in your word processor.
9. Practice saving and exporting a document in different formats.

10. Create a template for a recurring document like a meeting agenda or report.

Unit -III: Spreadsheet and Database Management

1. Create a basic spreadsheet and input some sample data.
2. Apply conditional formatting and data validation to a spreadsheet.
3. Create a chart and a pivot table to analyze the data in your spreadsheet.
4. Use a database tool to create tables, relationships, and queries.
5. Import data from an external source into your database.
6. Export data from your database to a spreadsheet.
7. Automate a simple task in your spreadsheet or database using macros or scripts.
8. Practice analyzing large sets of data in your spreadsheet.
9. Create a database query that requires multiple conditions.
10. Create a form for data entry in your database.

Unit -IV: Communication and Collaboration Tools; Security and Privacy in Office Automation

1. Create an email account and send a message.
2. Set up an email filter, rule, and signature.
3. Participate in an online meeting or web conference.
4. Use a collaboration tool to create shared calendars, task lists, or manage a project.
5. Research common security threats in office automation and summarize your findings.
6. Use a tool to encrypt a message or a file.
7. Create a guide for best practices in secure communication and collaboration.
8. Demonstrate how to securely share a file or document with others.
9. Create a mock phishing email and discuss how to identify and handle such threats.
10. Investigate a recent data breach related to office automation and present a case study.

Course Assessment (Full 50 Marks Internal Assessment)

Here are some potential assessments that could be used to evaluate understanding and practical skills for this course:

Unit -I: Introduction to Office Automation

1. **Paper/Report:** Submit a report on the history and evolution of office automation.
2. **Presentation:** Give a presentation on current trends in office automation.
3. **Case Study Evaluation:** Evaluate a case study on a company that has successfully implemented office automation.

Unit -II: Word Processing and Document Management

1. **Document Creation:** Create a document using advanced formatting techniques such as styles, templates, and themes.
2. **Mail Merge Assignment:** Perform a mail merge operation and submit the resulting documents.
3. **Collaborative Document Editing:** Participate in a collaborative document editing exercise and demonstrate the ability to review and revise the document.

Unit -III: Spreadsheet and Database Management

1. **Spreadsheet Assignment:** Create a complex spreadsheet that includes conditional formatting, data validation, charts, and pivot tables.

2. **Database Design:** Design a database with multiple related tables and demonstrate the ability to query the data.
3. **Data Import/Export:** Successfully import and export data between a spreadsheet and a database.

Unit -IV: Communication and Collaboration Tools; Security and Privacy in Office Automation

1. **Communication Exercise:** Demonstrate the use of advanced email features and participate in an online meeting or web conference.
2. **Collaboration Project:** Engage in a group project using advanced collaboration tools and submit the final output.
3. **Security and Privacy Assessment:** Create a presentation or report on security threats and best practices for secure communication and collaboration in the context of office automation.
4. **Encryption Exercise:** Demonstrate the ability to encrypt and decrypt a message or file.

Each of these assessments should be graded not only for the final output but also for the process used to create them. The idea is to evaluate the students' understanding and their ability to apply the concepts they've learned in practical situations. It's also crucial to provide clear criteria for each assignment so students know what is expected of them.

Course Code: CS-341T(B)	Course Title:- Critical Thinking
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: There are no prerequisites for this course	
Learning Objectives	
<ol style="list-style-type: none"> 1. To understand the concept of critical thinking and its significance in personal and professional life 2. To develop critical thinking skills like analysis, interpretation, evaluation, inference, and explanation 3. To apply critical thinking skills in decision-making and problem-solving 4. To exercise and improve the brain's ability to think critically 	
Learning Outcomes	
After Completion of the Course students will be able to	
<ol style="list-style-type: none"> 1. Develop critical thinking skills and apply them in various aspects of personal and professional life 2. Make informed decisions by analyzing information and evaluating options 3. Improve problem-solving skills by breaking down complex problems into smaller components 4. Enhance cognitive abilities to think critically and make logical decisions. 	
Unit -I: Introduction to Critical Thinking (10 Periods)	
Understanding the concept of critical thinking, Historical details of critical thinking, Thinkers who fashioned critical thinking of their time	
Unit -II: Developing Critical Thinking Skills (10 Periods)	

The process of critical thinking, Inductive and deductive reasoning, Difference between reading and thinking, Reason to Adopt Critical Thinking, How critical thinking solves problems

Unit -III: Improving Decision Making (10 Periods)

Getting logical thinking, Strategies to improve decision-making skills, Making better decisions

Unit -IV: Applying Critical Thinking (10 Periods)

Strategies to help improve critical thinking, Group decision-making skills, Applying questions in critical thinking, Exercising the brain

Unit-V: Test and Tutorials (05 Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. "Critical Thinking: Proven Strategies To Improve Decision Making Skills, Increase Intuition And Think Smarter" by Simon Bradley.

References:

1. "Thinking Critically" by John Chaffee (Oxford University Press India)
2. "Critical Thinking: An Introduction" by Alec Fisher (Cambridge University Press India)
3. "The Miniature Guide to Critical Thinking" by Richard Paul and Linda Elder (Foundation for Critical Thinking India)
4. "Asking the Right Questions: A Guide to Critical Thinking" by M. Neil Browne and Stuart M. Keeley (Pearson India)"Critical Thinking: Tools for Taking Charge of Your Learning and Your Life" by Richard Paul and Linda Elder (Pearson India)

E-Resources

1. <https://argumentful.com/16-best-free-online-critical-thinking-courses/>

Course Assessment (Full 50 Marks Internal Assessment)

To assess the skills acquired in a critical thinking course, you can use a combination of formative and summative assessment methods, including written assignments, discussions, group activities, quizzes, tests, and self-assessment. Here are some suggestions:

1. **Written Assignments:** Assign tasks that require students to analyze, evaluate, and synthesize information, such as essays, case studies, and reflections. These assignments can be graded based on predefined rubrics that outline expectations for clarity, depth, and logical reasoning.
2. **Discussions:** Organize in-class or online discussions in which students are required to critically analyze and evaluate different viewpoints, arguments, or evidence. Encourage students to ask probing questions and provide reasoned responses. Assess students' participation and the quality of their contributions.
3. **Group Activities:** Assign group projects or activities that require students to collaborate, analyze problems, and develop solutions using critical thinking skills. Evaluate the projects based on the quality of the work produced, as well as each student's participation and contribution to the group.
4. **Quizzes and Tests:** Create quizzes and tests that evaluate students' understanding of critical thinking concepts and their ability to apply these skills. Assessments can include multiple-choice questions, true/false questions, and short-answer questions. Quizzes can be administered throughout the course to gauge understanding, while tests can be used at

the end of each unit or at the end of the course to assess overall learning.

5. **Self-assessment:** Encourage students to self-assess their progress and skill development in critical thinking throughout the course. This can be done through reflective journaling, self-assessment checklists, or periodic self-evaluations.
6. **Peer Assessment:** Have students review and evaluate their peers' work, providing constructive feedback on areas for improvement. This can help students develop their own critical thinking skills and foster a collaborative learning environment.
7. **In-class Activities:** Conduct hands-on, in-class activities that allow students to practice their critical thinking skills in real-time. Observe how students analyze problems, evaluate evidence, and generate solutions, and provide feedback and support as needed.

By using a combination of these assessment methods, you can effectively evaluate students' skills and knowledge in critical thinking, ensuring that they have developed the necessary competencies for academic and professional success.

Here are sample questions for each of the suggested assessment methods:

1. Written Assignments:

Write an essay analyzing a controversial issue, discussing the main arguments on both sides, and presenting your own reasoned conclusion.

2. Discussions:

In a class discussion, debate the merits of implementing a new policy in a given context (e.g., a workplace, school, or government). Encourage students to ask probing questions and provide well-reasoned arguments.

3. Group Activities:

As a team, analyze a real-life case study involving a complex problem. Develop a solution using critical thinking skills and present your findings to the class.

4. Quizzes and Tests:

Multiple-choice question: Which of the following is an example of inductive reasoning?

- a) All dogs are mammals. Rover is a dog. Therefore, Rover is a mammal.
- b) Every time you eat peanuts, you have an allergic reaction. Therefore, you are allergic to peanuts.
- c) If it rains, the streets will be wet. The streets are wet. Therefore, it rained.
- d) A triangle has three sides. This shape has three sides. Therefore, this shape is a triangle.

5. **True/False** question: Critical thinking requires accepting arguments at face value without questioning the underlying assumptions or evidence.

6. Self-assessment:

Reflect on your growth in critical thinking skills throughout the course. Identify two areas where you have improved, and discuss one area where you still need to improve.

7. Peer Assessment:

Review a classmate's essay on a controversial issue. Provide feedback on the clarity, organization, and depth of their analysis, as well as the strength of their arguments. Suggest at least two specific improvements.

8. In-class Activities:

Participate in a group exercise where students are presented with a hypothetical scenario and must use critical thinking skills to evaluate the situation and make decisions. Observe and provide feedback on students' problem-solving and decision-making processes.

By incorporating these sample questions and activities into your assessments, you can effectively gauge students' understanding and mastery of critical thinking skills.



B. Sc. (Computer Science)

Semester - IV

Curriculum for semester IV

Course Code: CS-411T	Course Title:- Core Java
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: <ol style="list-style-type: none"> 1. Basic Programming Language Construct (Like looping and decision making) using C or C++ 2. Functions and Structures in C or C++ 	
Learning Objectives <ol style="list-style-type: none"> 1. To introduce students to the fundamental concepts of Java programming language. 2. To enable students to develop skills in writing and implementing exception handling in Java. 3. To introduce students to the concepts of constructors, wrapper classes, and string operations in Java. 4. To introduce students to the concepts of interfaces and threads in Java programming. 	
Learning Outcomes After Completion of the Course students will be able to <ol style="list-style-type: none"> 1. Understand the basic concepts of Java programming language. 2. Write and implement exception handling in Java. 3. Use constructors, wrapper classes, and string operations in Java programming. 4. Use interfaces and threads in Java programming to create multi-threaded applications. 	
Unit -I: Introduction to Java (10 Periods) Introduction of Java, History of Java, How Java is different from C++, JDK Tools, Class File, Java Bytecode, JVM, identifiers, Data types, Operators, Control Statements, loop, arrays, Inheritance in Java, Multilevel hierarchy, method overriding, Abstract classes, Final classes	
Unit -II: Package and Exception in Java (10 Periods) Defining, Implementing and Applying Packages, Importing Packages, Types of packages, User define package, Exception handling in Java, try, catch, throw, throws and finally, Uncaught Exceptions, Multiple catch, Java's Built-in Exception	
Unit -III: Constructor, Wrapper, String and StringBuffer Class in Java (10 Periods) Constructors, Various Types of Constructor, Role of Constructors in inheritance, Introduction to Wrapper Classes, String Operations in java, Immutability, Creating and Initializing Strings using methods of String and StringBuffer Class.	
Unit -IV: Interface and Threads in Java (10 Periods) Interface: Defining Interfaces, Abstract Methods in Interfaces, Implementing Interfaces, Extending Interfaces, Interface References, Default Methods in Interfaces, Static Methods in Interfaces, Constants in Interfaces Thread: Thread life cycle, Creating and implementing thread, multi-threaded programming, thread priorities, synchronization of thread, resuming and stopping Threads	
Unit-V: Test and Tutorials (05 Periods) In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).	

Textbook

1. "Programming with Java: A Primer" by E. Balagurusamy
2. "Core Java Volume I -- Fundamentals" by Cay S. Horstmann and Gary Cornell, published by Pearson Education India.

References:

1. "Java: The Complete Reference" by Herbert Schildt, published by McGraw Hill Education India.
2. Head First Java" by Kathy Sierra and Bert Bates
3. "Effective Java" by Joshua Bloch

E-Resources

1. Oracle Java Tutorials: <https://docs.oracle.com/en/java/javase/index.html>
2. Java Tutorial for Complete Beginners by John Purcell: <https://www.udemy.com/course/java-tutorial/>
3. Java Programming Basics by SoloLearn: <https://www.sololearn.com/learning/1060>

Course Code: CS-412T	Course Title: Computer Graphics
Total Credit:02	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: 1. Mathematics. 2. Good programming skills in C&C++ 3. Data Structures	
Learning Objectives <ul style="list-style-type: none">• Understanding how the various elements that like algebra, geometry, algorithms and data structures interact in the design of graphics.• To provides an idea on hardware system architecture for computer graphics.• To give idea about basic building blocks of multimedia	
Learning Outcomes After the completion of this course student should apply its real time application knowledge for <ul style="list-style-type: none">• Geometrical Transformations in 2-Dimensional and 3-Dimensional perspectives• Object representations• Surface detection procedures• Computer Animations	
Unit -I: Introduction to Computer Graphics: Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Input Devices for Operator Interaction, Active and Passive Graphics Devices, Display Technologies, Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays, Raster Refresh (Raster-Scan) Graphics Displays, Cathode Ray Tube Basics, Color CRT Raster Scan Basics, Video Basics, The Video Controller, Random-Scan Display Processor, LCD displays.	

Unit – II: Scan conversion:

Scan conversion: Digital Differential Analyzer (DDA) algorithm, Bresenham's Line drawing algorithm. Bresenham's method of Circle drawing, Midpoint Circle Algorithm, Midpoint Ellipse Algorithm, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles,

Unit – III: 2-D Geometrical transforms:

Two-Dimensional Transformations: Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation

Unit – IV: 3-D Three-Dimensional Transformations

Three-Dimensional Transformations: Three-Dimensional Scaling, Three-Dimensional Shearing, Three- Dimensional Rotation, Three-Dimensional Reflection, Three- Dimensional Translation, Multiple Transformation,

Introduction to animation: Design of animation sequence, general computer animation functions, raster animation

Unit-V: Test and Tutorials

In addition to CIA, Tutorial, Seminars, Assignment & case studies are to be given for building proficiency in the course. (Respective Course in-charge should maintain the records for the same).

TEXT BOOKS

1. "Computer Graphics C version", Donald Hearn and M. Pauline Baker, Pearson education.
2. "Computer Graphics Second edition", Zhigandxiang, Roy Plastock, Schaum's outlines, Tata Mc Graw hill edition.

Books Recommended:

1. J.D.Foley, A.Van Dan, Feiner, Hughes Computer Graphics Principles & Practice 2nd edition Publication Addison Wesley 1990.
2. D.Hearn, Baker: Computer Graphics, Prentice Hall of India 2008.
3. D.F.Rogers Procedural Elements for Computer Graphics, McGraw Hill 1997.
4. D.F.Rogers, Adams Mathematical Elements for Computer Graphics, McGraw Hill 2nd edition 1989.

NPTEL Video :

1. <http://nptel.ac.in/courses/106106090/#>

Free E-Books

1. <https://www.pdfdrive.com/computer-graphics-books.html>
2. <https://www.pdfdrive.com/introduction-to-computer-graphics-e34322358.html>

Course Code: CS-413P	Course Title: Lab Course (based on CS-411T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-411** (The teacher can add three practical examples based on each unit as per the choice and feasibility, the below provided list is a sample list of experiments)

1. Creating a simple Java program to print "Hello World"
2. Implementing basic control statements such as if, else, switch, and loops
3. Creating and using arrays in Java
4. Implementing inheritance in Java with multilevel hierarchy
5. Overriding methods in Java and using final and abstract classes
6. Implementing user-defined packages and importing them
7. Implementing exception handling in Java with try, catch, throw, and finally blocks
8. Creating and using wrapper classes in Java
9. String operations in Java such as concatenation, substring, and length
10. Creating and using interfaces in Java
11. Implementing multi-threaded programming in Java
12. Implementing thread synchronization in Java
13. Creating and using constructor methods in Java
14. Using String and StringBuffer classes in Java
15. Creating a Java program that combines multiple concepts such as inheritance, interfaces, and exception handling.

Course Code: CS-414P	Course Title: Lab Course(Lab based on CS-412T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-412T** (Computer Graphics)

This practical can be implemented in C or C++ programming language.

1. Study and enlist the basic functions used for graphics in C / C++ language. Give an example for each of them.
2. Draw a co-ordinate axis at the center of the screen.
3. Divide your screen into four region, draw circle, rectangle, ellipse and half ellipse in each region with appropriate message.
4. Draw a simple hut on the screen.
5. Draw the following basic shapes in the center of the screen:
 - i. Circle
 - ii. Rectangle

- iii. Square
 - iv. Concentric Circles
 - v. Ellipse
 - vi. Line
6. Develop the program for DDA Line drawing algorithm
 7. Develop the program for Bresenham's Line drawing algorithm.
 8. Develop the program for the mid-point circle drawing algorithm.
 9. Develop the program for the mid-point ellipse drawing algorithm
 10. Write a program to implement 2D scaling
 11. Write a program to perform 2D translation
 12. Perform 2D Rotation on a given object.
 13. Program to create a house like figure and perform the following operations.
 - i. Scaling about the origin followed by translation.
 - ii. Scaling with reference to an arbitrary point.
 - iii. Reflect about the line $y = mx + c$.
 14. Develop a simple text screen saver using graphics functions
 15. Perform smiling face animation using graphic functions.

Course Code: CS-421T	Course Title: Basics of Android OS
Total Credit:02	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: 1. Basic Knowledge of Programming, Concepts of OOPS	
Learning Objectives	
<ol style="list-style-type: none"> 1. Describe Platforms on which Android operating system will run. Install Android studio 2. Understand the fundamentals of Android Architecture 3. Create simple application which runs under Android Operating system 4. Understand the UI components 5. Explain event handling and create style sheets 	
Learning Outcomes	
On successful completion of the course, students will be able to do following:	
<ol style="list-style-type: none"> 1. Student should perfect in the android operating system and its real time application development. 	
Unit-I:	
Environment Setup: Setup Java Development Kit (JDK), Android SDK, Android Development Tools (ADT) Plugin, Create Android Virtual Device,	
Architecture: Linux kernel, Libraries, Android Runtime, Application Framework, Applications.	
Application Components	
Activities, Services, Broadcast Receivers, Content Providers, Additional Components, Create Android	

Application, Anatomy of Android Application, The Main Activity File, The Manifest File, The Strings File, The R File, The Layout File, Running the Application.

Unit-II:

Resources Organizing & Accessing: Alternative Resources, Accessing Resources

UI Layouts Android Layout Types: Relative Layout Attributes, Grid View Attributes, Sub-Activity, Layout Attributes, View Identification,

Android UI Controls: TextView Attributes, EditText Attributes, AutoComplete Text View Attributes, Button Attributes, ImageButton Attributes, CheckBox Attributes, ToggleButton Attributes, RadioButton Attributes, RadioGroup Attributes.

Unit-III:

Intents and Filters: Intent Objects, Action, Android Intent Standard Actions, Data, Category, Extras, Flags, Component Name, Types of Intents: Explicit Intents, Implicit Intents.

Fragments: Fragment Life Cycle, Creating new Fragments, Fragment States, Adding Fragments to activities.

Unit-IV:

Event Handling: Event Listeners & Event Handlers, Event Listeners Registration, Styles and Themes, Defining Styles, Using Styles, Style Inheritance, Android Themes, Default Styles & Themes, Custom Components, Creating a Simple Custom Components.

Unit-V: Test and Tutorials

In addition to CIA, Tutorial, Seminars, Assignment & case studies are to be given for building proficiency in the course. (Respective Course in-charge should maintain the records for the same).

Textbook

1. Android Application Development (O'Reilly)
2. Head First Android Development: A Brain-Friendly GuideBook by David Griffiths and Dawn Griffiths

Online Resources

1. <https://developer.android.com/guide>
2. <https://www.tutorialspoint.com/android/index.htm>

Reference Book

1. Learn Android App Development by Wallace Jackson
2. Android App Development for Dummies, 3ed by Michael Burton

Course Code: CA-422T	Course Title:- Computer Networks
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: Basic knowledge of computer systems and programming languages.	
Learning Objectives <ol style="list-style-type: none"> 1. Understand the fundamental concepts of computer networks and their applications. 2. Gain knowledge of the layered network architecture and various network protocols and services. 3. Understand the functions and operation of various network layers, including the physical, data link, network, transport, and application layers. 4. Gain knowledge of network security threats and vulnerabilities, as well as various security protocols and encryption techniques. 	
Learning Outcomes After Completion of the Course students will be able to <ol style="list-style-type: none"> 1. Design and implement computer networks. 2. Analyze and troubleshoot network-related problems. 3. Understand various network protocols and services. 4. Understand network security threats and vulnerabilities, as well as various security protocols and encryption techniques. 	
Unit -I: Introduction to Computer Networks (10 Periods) Overview of computer networks and their applications, Network topologies and architectures, Layered network architecture and the OSI reference model, Network protocols and services	
Unit -II: Physical Layer and Data Link Layer (10 Periods) Overview of the physical layer and its functions, Transmission media and their characteristics, Data encoding and modulation techniques, Error detection and correction, Data link layer and its functions, Framing, flow control, and error control in data link layer protocols.	
Unit -III: Network Layer and Transport Layer (10 Periods) Overview of the network layer and its functions, Routing algorithms and protocols, IPv4 and IPv6 addressing and routing, Transport layer and its functions, Reliable data transfer and flow control, TCP and UDP protocols.	
Unit -IV: Application Layer and Security (10 Periods) Overview of the application layer and its functions, Client-server and peer-to-peer architectures, Common application layer protocols (HTTP, FTP, SMTP, DNS), Network security threats and vulnerabilities, Cryptography and encryption techniques, Security protocols (SSL/TLS, IPsec, VPN).	
Unit-V: Test and Tutorials (05 Periods) In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency	

in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall
2. "Data Communications and Networking" by Behrouz A. Forouzan

References:

1. "TCP/IP Protocol Suite" by Behrouz A. Forouzan
2. "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross
3. "Networking Essentials" by Jeffrey S. Beasley and Piyasat Nilkaew

E-Resources

1. Cisco Networking Academy: <https://www.netacad.com/courses/networking>
2. Computer Networking: Principles, Protocols, and Practice (Open Textbook): <https://www.computer-networking.info/>
3. Coursera Networking Courses: <https://www.coursera.org/courses?query=networking>
4. MIT OpenCourseWare: Computer Networks: <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-829-computer-networks-fall-2002/>

Course Code: CS-423P	Course Title:Lab Course (Lab based on CS-421T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course CS-421T (The teacher can add three practical examples based on each unit as per the choice and feasibility, the below provided list is a sample list of experiments)

List of Practical (Minimum 10)

1. Practical No.1: Installing "Android Studio IDE" and "Android SDK"
2. Working with Linear Layout and UI components in Android
3. Working with Relative Layout and UI components in Android
4. Working with Table Layout and UI components in Android
5. Working with UI components (TextView, EditText, RadioButton, ToggleButton, CheckBox, RatingBar, AutocompleteTextView)
6. Create Android Application to demonstrate button click event
7. Create Android Application to demonstrate RadioButton checked event
8. Create Android Application to demonstrate ToggleButton clicked event and change attributes of Layout/UI components
9. Create Android Application to demonstrate basic calculator activity_main.xml
10. Design Android Application components using style sheet.

Course Code: CS-424P	Course Title: Lab Course (based on CS-422T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course CS-422T (The teacher can add three practical examples based on each unit as per the choice and feasibility, the below provided list is a sample list of experiments)

Unit 1: Introduction to Computer Networks

1. Setting up a simple LAN network using Ethernet cables and a switch
2. Configuring network settings on a Windows or Linux computer
3. Implementing network addressing using IP addresses and subnet masks
4. Setting up a wireless network using a wireless router
5. Capturing and analyzing network traffic using Wireshark

Unit 2: Physical Layer and Data Link Layer

1. Measuring and comparing the performance of different transmission media, such as twisted-pair cables, fiber optic cables, and wireless signals
2. Implementing and comparing different error detection and correction algorithms, such as parity check, CRC, and Hamming code
3. Setting up a simple data link layer protocol using Python or Java
4. Implementing and analyzing different flow control algorithms, such as stop-and-wait and sliding window
5. Building and testing a simple network using virtual machines and virtual network adapters

Unit 3: Network Layer and Transport Layer

1. Implementing and comparing different routing algorithms, such as distance vector and link-state
2. Setting up a simple router using Linux or Cisco routers and configuring routing tables
3. Implementing and testing different network layer protocols, such as ARP and ICMP
4. Implementing and analyzing different transport layer protocols, such as TCP and UDP, using packet sniffing tools
5. Building and testing a simple client-server application using TCP or UDP sockets

Unit 4: Application Layer and Security

1. Implementing and testing simple HTTP and FTP clients and servers
2. Configuring and testing a simple mail server using SMTP and POP3 protocols
3. Implementing and testing a simple DNS server using Python or Java
4. Analyzing network traffic to identify security threats and vulnerabilities using network analysis tools
5. Implementing and testing different security protocols, such as SSL/TLS, IPsec, and VPN, using virtual private networks and security appliances.

Course Code: CS-431T	Course Title: Data Analytics
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites:	
1. Basic Knowledge of Microsoft Excel or Google sheets	
Course Objectives	
<ol style="list-style-type: none"> 1. Develop Excel proficiency for data analytics 2. Understand data analysis concepts and techniques 3. Use advanced Excel features for data visualization 4. Learn practical skills for data-driven decision-making 	
Course Outcomes	
Upon Completion of the course the students will be	
<ol style="list-style-type: none"> 1. Able to analyze data using Excel. 2. Able to effectively visualize data using advanced charting techniques 3. Apply statistical analysis techniques in business scenarios 4. Able to make data-driven decisions using advance analysis techniques. 	
Unit-I: Understanding Data Analytics and Excel, Data import & Pre-processing: (10 Periods)	
1: Introduction to Data Analytics and Excel Basics	
<ul style="list-style-type: none"> • Understanding the importance of data analytics. • Introduction to Excel as a data analytics tool • Familiarization with the Excel interface 	
2: Data Types, Formats, and Basic Excel Functions	
<ul style="list-style-type: none"> • Learning about data types and formats in Excel • Introduction to Excel functions and formulas • Practice with basic functions and formulas 	
3: Data Import Techniques	
<ul style="list-style-type: none"> • Importing data from various sources • Introduction to Excel's data import tools • Hands-on practice with importing data 	
4: Data Cleaning and Transformation	
<ul style="list-style-type: none"> • Understanding the need for data cleaning • Techniques for data transformation and normalization • Hands-on practice with data cleaning and transformation 	
5: Handling Missing Values in Excel	
<ul style="list-style-type: none"> • Identifying and understanding missing values • Methods for handling missing values in Excel • Hands-on practice with missing value management 	
6: Data Formatting in Excel	
<ul style="list-style-type: none"> • Introduction to data formatting in Excel • Customizing cell formats for better data presentation • Hands-on practice with data formatting 	
7: Conditional Formatting in Excel	
<ul style="list-style-type: none"> • Understanding conditional formatting • Applying conditional formatting rules to improve data visualization • Hands-on practice with conditional formatting 	
8: Advanced Excel Functions and Formulas	
<ul style="list-style-type: none"> • Introduction to advanced Excel functions and formulas • Hands-on practice with advanced functions and formulas 	
9: Data Analytics Techniques in Excel	
<ul style="list-style-type: none"> • Understanding various data analytics techniques • Applying Excel functions and tools for data analysis 	

- Hands-on practice with data analytics techniques in Excel

10: Unit 1: Project and Review

- Applying the learned concepts to a real-world data analytics project
- Review of key concepts and techniques
- Presentation and discussion of final projects

Unit-II: Descriptive Statistics and Data Visualization, Data Analysis Techniques: (10 Periods)

1: Basic Statistical Functions

- Introduction to descriptive statistics
- Using basic statistical functions in Excel: COUNT(), SUM(), AVERAGE(), MEDIAN(), MODE(), MIN(), MAX(), STDEV()
- Hands-on practice with basic statistical functions

2: Frequency Distributions and Histograms

- Understanding frequency distributions and their importance
- Creating frequency distributions and histograms in Excel
- Hands-on practice with frequency distributions and histograms

3: Pivot Tables and Pivot Charts

- Introduction to PivotTables and PivotCharts
- Creating and customizing PivotTables and PivotCharts for data summarization
- Hands-on practice with PivotTables and PivotCharts

4: Basic Excel Charts for Data Visualization

- Introduction to basic Excel chart types: column, bar, line, pie, and area charts
- Creating and customizing basic Excel charts
- Hands-on practice with basic chart types

5: Advanced Chart Types and Customization

- Exploring advanced Excel chart types: scatter, bubble, radar, waterfall, and treemap charts
- Customizing chart elements and formatting for effective data visualization
- Hands-on practice with advanced chart types

6: Sorting and Filtering Data

- Introduction to sorting and filtering data in Excel
- Using sorting and filtering tools for data organization and analysis
- Hands-on practice with sorting and filtering

7: Data Validation and Data Auditing

- Understanding data validation and its importance
- Implementing data validation rules in Excel
- Introduction to data auditing tools and techniques
- Hands-on practice with data validation and auditing

8: Advanced Excel Functions for Data Analysis

- Introduction to advanced Excel functions: VLOOKUP(), HLOOKUP(), INDEX(), MATCH(), COUNTIF(), SUMIF()
- Hands-on practice with advanced functions for data analysis

9: What-If Analysis: Goal Seek

- Understanding Goal Seek and its applications
- Using Goal Seek to find input values that achieve a specific goal
- Hands-on practice with Goal Seek

10: What-If Analysis: Data Tables and Scenario Manager

- Introduction to Data Tables and Scenario Manager for what-if analysis
- Creating one-variable and two-variable data tables
- Using Scenario Manager to analyze different scenarios and their impact
- Hands-on practice with Data Tables and Scenario Manager

11: Unit 2: Project and Review

- Applying the learned concepts to a real-world data analytics project
- Review of key concepts and techniques
- Presentation and discussion of final projects

Unit-III: Working with Time Series Data & Regression Analysis: (10 Periods)

1: Introduction to Time Series Data

- Understanding time series data and its importance
- Working with time series data in Excel: date and time functions
- Hands-on practice with time series data manipulation

2: Trend Analysis and Forecasting

- Identifying trends and patterns in time series data
- Introduction to time series forecasting
- Forecasting techniques in Excel: linear and polynomial trendlines
- Hands-on practice with trend analysis and forecasting

3: Smoothing Techniques: Moving Averages

- Introduction to moving averages as a smoothing technique
- Calculating simple, weighted, and exponential moving averages in Excel
- Hands-on practice with moving averages for trend analysis

4: Smoothing Techniques: Exponential Smoothing

- Understanding exponential smoothing and its applications
- Implementing exponential smoothing in Excel using the "Forecast Sheet" feature
- Hands-on practice with exponential smoothing for forecasting

5: Simple Linear Regression

- Introduction to simple linear regression analysis
- Using Excel's Data Analysis ToolPak to perform simple linear regression
- Interpreting regression output and understanding coefficient estimates
- Hands-on practice with simple linear regression

6: Multiple Linear Regression

- Introduction to multiple linear regression analysis
- Performing multiple linear regression using Excel's Data Analysis ToolPak
- Interpreting multiple regression output and understanding coefficient estimates
- Hands-on practice with multiple linear regression

7: Model Diagnostics and Validation

- Assessing the quality of regression models: R-squared, adjusted R-squared, and standard error
- Testing for assumptions: normality, linearity, multicollinearity, and homoscedasticity
- Cross-validation and model selection techniques
- Hands-on practice with model diagnostics and validation

8: Nonlinear Regression Models

- Introduction to nonlinear regression models
- Implementing nonlinear regression models in Excel using the Solver add-in
- Hands-on practice with nonlinear regression

9: Time Series Decomposition

- Understanding the components of time series data: trend, seasonality, and noise
- Decomposing time series data in Excel using moving averages and seasonal indices
- Hands-on practice with time series decomposition

10: Advanced Time Series Forecasting Techniques

- Introduction to advanced time series forecasting techniques: autoregressive (AR) and moving average (MA) models
- Implementing advanced forecasting techniques in Excel using custom formulas and add-ins
- Hands-on practice with advanced time series forecasting techniques

11: Unit 3: Project and Review

- Applying the learned concepts to a real-world data analytics project
- Review of key concepts and techniques
- Presentation and discussion of final projects

Unit-IV: Hypothesis Testing, Confidence Intervals And Excel Add-ins for Data Analytics: (10 Periods)

1: Hypothesis Testing Basics

- Introduction to hypothesis testing and its importance

- Understanding null and alternative hypotheses
 - Types of hypothesis tests: one-tailed and two-tailed tests
 - Hands-on practice with hypothesis testing in Excel
- 2: Confidence Intervals
- Understanding confidence intervals and their interpretation
 - Calculating confidence intervals for means and proportions in Excel
 - Hands-on practice with constructing confidence intervals
- 3: T-Tests and Z-Tests
- Introduction to t-tests and z-tests
 - Performing one-sample, two-sample, and paired t-tests in Excel using the Data Analysis ToolPak
 - Conducting z-tests in Excel using custom formulas
 - Hands-on practice with t-tests and z-tests
- 4: Chi-Square Tests and ANOVA
- Introduction to chi-square tests for independence and goodness-of-fit
 - Performing chi-square tests in Excel using the Data Analysis ToolPak or custom formulas
 - Introduction to Analysis of Variance (ANOVA) for comparing multiple means
 - Conducting one-way and two-way ANOVA in Excel using the Data Analysis ToolPak
 - Hands-on practice with chi-square tests and ANOVA
- 5: Excel Analysis ToolPak
- Introduction to the Excel Analysis ToolPak and its features
 - Using the ToolPak for statistical analysis: t-tests, ANOVA, correlation, and regression
 - Hands-on practice with the Excel Analysis ToolPak
- 6: Power Query for Data Transformation
- Introduction to Power Query and its applications
 - Importing, cleaning, and transforming data using Power Query
 - Merging and appending queries to combine data from multiple sources
 - Hands-on practice with Power Query for data transformation
- 7: Power Pivot for Data Modeling
- Introduction to Power Pivot and data modeling in Excel
 - Creating and managing data models using Power Pivot
 - Working with calculated columns and measures using DAX (Data Analysis Expressions)
 - Hands-on practice with Power Pivot for data modeling
- 8: Power Map for Geospatial Data Visualization
- Introduction to Power Map (3D Maps) for geospatial data visualization
 - Creating interactive, 3D geospatial visualizations using Power Map
 - Customizing map layers, chart types, and visual elements
 - Hands-on practice with Power Map for geospatial data visualization
- 9: Advanced Hypothesis Testing Techniques
- Introduction to advanced hypothesis testing techniques: F-tests, Mann-Whitney U test, and Kruskal-Wallis test
 - Implementing advanced hypothesis tests in Excel using custom formulas or third-party add-ins
 - Hands-on practice with advanced hypothesis testing techniques
- 10: Data Analytics Project and Review
- Applying the learned concepts to a real-world data analytics project involving hypothesis testing, confidence intervals, and Excel add-ins
 - Review of key concepts and techniques covered in the chapters
 - Presentation and discussion of final projects

Unit-V: Test and Tutorials (05 Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. Mastering Data Analytics (Using Excel), By Dr. Nazneen Akhter & Prof. Bharti Gawali, ISBN 978-93-5542-402-0 (Shroff Publishers.)

References:

- 1. Data Analysis with Excel Paperback – 1 January 2019 by Manish Nigam.
- 2. Microsoft Excel Data Analysis and Business Modelling by Wayne Winston

E-Resources

- 1. Microsoft Excel Help Center: This is a comprehensive resource for all things in Excel, including tutorials, how-to guides, and troubleshooting tips.
- 2. Excel Easy: A free online tutorial website that covers all the basics of Excel and includes step-by-step guides for common data analysis tasks.

Course Code: CS-432T	Course Title:- Open-Source Web Application Development
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: Basic knowledge of HTML, CSS, and JavaScript	
Learning Objectives	
<ul style="list-style-type: none"> 1. Install and configure MySQL and Apache, and write basic PHP code to interact with them. 2. Develop an understanding of PHP syntax, data types, and control structures, and how to work with forms, cookies, and files. 3. Gain proficiency in advanced PHP topics such as functions, arrays, objects, and strings, dates, and time. 4. Build dynamic web applications using PHP and integrate them with MySQL and Apache. 	
Learning Outcomes	
After Completion of the Course students will be able to	
<ul style="list-style-type: none"> 1. Install and configure MySQL and Apache, and write basic PHP code to interact with them. 2. Write PHP scripts to handle form submissions, set and delete cookies, and interact with files and directories. 3. Develop an understanding of advanced PHP topics, including functions, arrays, objects, and strings, dates, and time. 4. Build dynamic web applications using PHP and integrate them with MySQL and Apache. 	
Unit -I: Introduction to Web Development with MySQL and Apache (10 Periods)	
Overview of web development and its components (HTML, CSS, JavaScript), Introduction to MySQL and Apache, Installing MySQL and Apache on a local machine, Basic security guidelines for MySQL and Apache.	
Unit -II: PHP Basics (10 Periods)	
Functions in PHP: meaning, calling, defining, and testing for existence, Arrays in PHP: creating and using arrays, and array-related functions, Objects in PHP: creating an object and object inheritance, Working with strings, dates, and time: formatting strings, using date/time functions, and other related functions.	
Unit -III: Advanced PHP Topics (10 Periods)	

PHP with AJAX: Introducing Ajax-Ajax Basics-PHP and Ajax-Database Driven Ajax. PHP with SEO: Basic SEO-Provocative SE Friendly URLs-Duplicate Content- CMS: Word press Creating an SE-Friendly Blog.

Unit -IV: Web Forms, Cookies, and File Handling (10 Periods)

Creating a simple input form and accessing form input with user-defined arrays, Working with HTML and PHP code on a single page, using hidden fields to save state, and redirecting the user, Introduction to cookies, setting and deleting cookies with PHP, and an overview of session functions, Working with files and directories: including files with include(), creating, deleting, opening, and validating files.

Unit-V: Test and Tutorials (05 Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. "PHP and MySQL Web Development" by Luke Welling and Laura Thomson
2. "Learning PHP, MySQL & JavaScript" by Robin Nixon
3. "Web Database Applications with PHP & MySQL" by Hugh E. Williams and David Lane
4. "PHP, MySQL, JavaScript & HTML5 All-in-One For Dummies" by Steve Suehring, Janet Valade, and Tricia Ballard
5. "Head First PHP & MySQL" by Lynn Beighley and Michael Morrison

References:

1. "PHP for the Web: Visual QuickStart Guide" by Larry Ullman
2. "PHP and MySQL: Novice to Ninja" by Kevin Yank
3. "Modern PHP: New Features and Good Practices" by Josh Lockhart

E-Resources

1. W3Schools (<https://www.w3schools.com/>): A popular website that offers tutorials and references on various web development technologies including HTML, CSS, JavaScript, PHP, MySQL, and more.
2. Mozilla Developer Network (<https://developer.mozilla.org/>): A comprehensive resource for web developers, offering documentation and tutorials on HTML, CSS, JavaScript, and other web technologies.
3. FreeCodeCamp (<https://www.freecodecamp.org/>): A non-profit organization that offers a free and interactive online platform to learn web development, including HTML, CSS, JavaScript, and more.
4. Codecademy (<https://www.codecademy.com/>): An online learning platform that offers interactive coding courses on various web development technologies including HTML, CSS, JavaScript, PHP, and more.

Course Code: CS-433P	Course Title: Practical Based on CS-431T
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-431T**

The following Experiments can be performed in Microsoft Excel and/or Google Sheets

1. Clean and prepare a messy dataset for analysis using Excel's data cleaning tools.
2. Use Excel's pivot tables and charts to explore and visualize data from a large dataset.
3. Use Excel's conditional formatting to highlight important data trends and outliers.
4. Use Excel's charting tools to create a scatter plot and identify correlation between two variables.
5. Use Excel's data filtering and sorting tools to explore a large dataset.
6. Use Excel's pivot tables and charts to create a dashboard that summarizes key metrics.
7. Use Excel's text-to-columns feature to split data in a single column into multiple columns.
8. Use Excel's remove duplicates feature to identify and remove duplicate entries in a dataset.
9. Use Excel's fill handle to quickly fill in missing data in a dataset.
10. Use Excel's SUMIFS function to sum data based on multiple criteria.
11. Use Excel's COUNTIF function to count data based on a specific condition.
12. Use Excel's AVERAGEIF function to calculate the average of data that meets a specific criterion.
13. Use Excel's pivot tables to calculate total sales by region and product category.
14. Use Excel's pivot tables to calculate the average order value by customer segment.
15. Use Excel's pivot tables to calculate the top selling products by region.
16. Use Excel's line chart to plot the trend of sales over time.
17. Use Excel's bar chart to compare sales across different product categories.
18. Use Excel's pie chart to visualize the percentage breakdown of sales by region.
19. Use Excel's combination chart to plot multiple data series on a single chart.
20. Use Excel's waterfall chart to visualize the contribution of each factor to a total value.
21. Use Excel's heat map chart to visualize the correlation between multiple variables.
22. Use Excel's t-test function to compare the means of two different datasets.
23. Use Excel's regression analysis tool to build a linear regression model.
24. Use Excel's ANOVA function to compare the means of three or more datasets.

Course Code: CS-434P	Course Title: Lab Course (based on CS-432T)
Total Credit: 1.5	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	

Sample List of experiments to be carried out based on the course **CS-432T (The teacher can add three practical examples based on each unit as per the choice and feasibility, the below provided list is a sample list of experiments)**

Unit 1: Introduction to Web Development with MySQL and Apache

1. Installing MySQL and Apache on a local machine
2. Configuring Apache server settings
3. Creating a basic HTML page and displaying it in a web browser
4. Connecting to MySQL database and creating a new database
5. Creating tables and inserting data into a MySQL database
6. Displaying data from a MySQL database on a web page
7. Configuring basic security settings for Apache and MySQL

8. Creating a login page with authentication using MySQL
9. Creating a registration form and storing user data in a MySQL database
10. Using Apache to serve static files like images and videos
11. Configuring Apache to work with PHP files
12. Creating a simple PHP script to display information from a MySQL database
13. Understanding and modifying the PHP configuration file (php.ini)
14. Testing web applications with Apache and MySQL
15. Troubleshooting common Apache and MySQL errors

Unit 2: PHP Basics

1. Creating a basic PHP script and displaying output in a web browser
2. Using variables and data types in PHP
3. Creating and using arrays in PHP
4. Working with strings and manipulating text in PHP
5. Using control structures like if/else statements and loops in PHP
6. Creating and calling functions in PHP
7. Creating and manipulating objects in PHP
8. Handling errors and exceptions in PHP
9. Working with dates and times in PHP
10. Reading and writing files in PHP
11. Uploading and handling files with PHP
12. Creating a simple login system with PHP
13. Using PHP to send emails
14. Understanding and working with sessions in PHP
15. Building a simple shopping cart with PHP and MySQL

Unit 3: Advanced PHP Topics

1. Creating and working with multidimensional arrays in PHP
2. Using PHP to work with JSON data
3. Understanding and using regular expressions in PHP
4. Using PHP to work with XML data
5. Creating and using namespaces in PHP
6. Implementing and using traits in PHP
7. Working with magic methods and properties in PHP
8. Understanding and using design patterns in PHP
9. Implementing a simple MVC (Model-View-Controller) architecture in PHP
10. Creating and using custom PHP extensions
11. Using PHP to work with databases other than MySQL
12. Creating a RESTful API with PHP
13. Using PHP to work with web services like SOAP and REST
14. Creating and using PHP libraries and frameworks
15. Debugging and profiling PHP code

Unit 4: Web Forms, Cookies, and File Handling

1. Creating a simple HTML form and processing the form data with PHP
2. Using PHP to handle user input validation and sanitization
3. Creating a file upload form and processing uploaded files with PHP
4. Using PHP to handle and manipulate images
5. Creating and using cookies in PHP
6. Implementing basic user authentication with cookies in PHP
7. Understanding and using session variables in PHP

8. Implementing advanced user authentication with sessions in PHP
9. Handling file input and output with PHP
10. Using PHP to work with ZIP files and archives
11. Implementing basic file encryption and decryption with PHP
12. Creating a simple file sharing system with PHP and MySQL
13. Working with directories and file permissions in PHP
14. Using PHP to work with remote files and resources Implementing a basic caching system with PHP

Course Code: CS-441T(C)	Course Title:- Basic Python Programing
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: Programing Methodology and Basic Programing Knowledge	
Learning Objectives	
<ol style="list-style-type: none"> 1. Enable students to understand Python's syntax and set up their programming environment. 2. Teach core programming concepts like variables, data types, control structures, and data structures. 3. Educate students about Python-specific concepts like functions, classes, exception handling, and file operations. 4. Introduce students to Python libraries and advanced topics like list comprehensions, lambda functions, and regular expressions. 	
Learning Outcomes	
After Completion of the Course students will be able to	
<ol style="list-style-type: none"> 1. Students will be able to comfortably use Python for programming tasks. 2. Students will develop strong problem-solving skills applicable to computational and real-world scenarios. 3. Students will be able to design and implement Python programs using OOP principles. 4. Students will be adept at using key Python libraries for numerical computation, data manipulation, and visualization. 	
Unit -I: Introduction to Python and Basics of Programming (10 Periods)	
<ol style="list-style-type: none"> 1. Introduction to Programming: Why learn Programming? 2. What is Python and Why Python? 3. Installing Python and setting up the development environment. 4. Understanding Python syntax and basic data types. 5. Variables and Operators in Python. 6. Conditional Statements: If, Else, Elif. 7. Looping Statements: While loop and For loop. 8. Python Data Structures: Lists, Tuples. 9. Python Data Structures: Sets, Dictionaries. 10. Practice and Review Session. 	
Unit -II: Python Functions, File I/O, and Exceptions (10 Periods)	
<ol style="list-style-type: none"> 1. Introduction to Functions in Python. 2. Defining Functions and Calling them. 	

3. Function Parameters, Return Values, and Scope.
4. Introduction to Python Modules and Packages.
5. File Operations: Opening, reading, writing, and closing files.
6. Introduction to Exceptions and Error Handling.
7. Try, Except, Else, Finally blocks.
8. Raising and catching exceptions.
9. Introduction to the 'with' statement for simplified File I/O and exception handling.
10. Practice and Review Session.

Unit -III: Object-Oriented Programming in Python (10 Periods)

1. Understanding the concept of Object-Oriented Programming.
2. Classes and Objects in Python.
3. The `__init__` method, class variables, and instance variables.
4. Inheritance in Python.
5. Overriding and Overloading methods.
6. Polymorphism in Python.
7. Encapsulation: Private methods and name mangling.
8. Abstract Classes and Interfaces.
9. Exception Handling in OOP.
10. Practice and Review Session.

Unit -IV: Libraries and Advanced Topics (10 Periods)

1. Introduction to Python Libraries.
2. Introduction to NumPy and its applications.
3. Introduction to Pandas for data manipulation.
4. Introduction to Matplotlib for data visualization.
5. Understanding List Comprehensions.
6. Introduction to Lambda functions and Map, Reduce, Filter functions.
7. Generators, Decorators, and Context Managers in Python.
8. Overview of Regular Expressions in Python.
9. Introduction to Python's Standard Library.
10. Practice and Review Session.

Unit-V: Test and Tutorials (05 Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. "Python Crash Course: A Hands-On, Project-Based Introduction to Programming" by Eric Matthes
2. "Learn Python the Hard Way" by Zed A. Shaw
3. "Automate the Boring Stuff with Python" by Al Sweigart

References:

1. "Fluent Python: Clear, Concise, and Effective Programming" by Luciano Ramalho
2. "Effective Python: 90 Specific Ways to Write Better Python" by Brett Slatkin
3. "Python Cookbook: Recipes for Mastering Python 3" by David Beazley and Brian K. Jones

E-Resources

1. Official Python Documentation: The official Python documentation is a comprehensive resource that covers everything from basic syntax to advanced topics. [Link](#)

2. Codecademy Python Course: This interactive course offers hands-on experience and covers a wide variety of Python topics. [Link](#)
3. LeetCode Python Problems: Practicing problems on LeetCode can help solidify programming concepts and improve problem-solving skills. [Link](#)

Some practical exercises for each unit:

Unit 1: Introduction to Python and Basics of Programming

1. Write a Python program to print "Hello, World!"
2. Write a Python program to perform basic arithmetic operations.
3. Write a program to swap two variables.
4. Write a Python program to check if a number is even or odd using conditional statements.
5. Write a Python program to print the Fibonacci sequence up to n terms.
6. Write a Python program to find the factorial of a number using a loop.
7. Write a Python program that sorts a list of numbers in ascending order.
8. Write a Python program to find the largest number in a list.
9. Write a Python program to remove duplicates from a list.
10. Write a Python program that counts the frequency of elements in a list using a dictionary.

Unit 2: Python Functions, File I/O, and Exceptions

1. Write a Python function that checks if a number is a palindrome.
2. Write a Python function that accepts a string and calculates the number of uppercase and lowercase letters.
3. Create a Python module with multiple functions and then import it in another Python program.
4. Write a Python program that reads a file, prints the content of the file, and counts the number of lines in the file.
5. Write a Python program that writes data into a file.
6. Write a Python program that copies the content from one file to another.
7. Write a Python program that handles a divide by zero exception.
8. Write a Python program that uses try, except, else, and finally blocks.
9. Write a Python program that raises a type error when the wrong data type is provided as input.
10. Write a Python program that uses the 'with' statement to read a file.

Unit 3: Object-Oriented Programming in Python

1. Write a Python class that represents a rectangle, including methods for calculating its area and perimeter.
2. Write a Python class with a method that checks if a string is a palindrome.
3. Write a Python class with an `__init__` method, class variables, and instance variables.
4. Write a Python program to demonstrate the concept of inheritance.
5. Write a Python program to demonstrate the concept of method overriding.
6. Write a Python program to demonstrate the concept of polymorphism.
7. Write a Python program to demonstrate the concept of encapsulation.
8. Write a Python program to demonstrate the concept of abstraction.
9. Write a Python program to handle an exception in a method of a class.
10. Write a Python program to demonstrate multiple inheritance.

Unit 4: Libraries and Advanced Topics

1. Write a Python program using NumPy to create a 2D array and perform basic operations.

2. Write a Python program using Pandas to read a CSV file and perform basic data manipulation.
3. Write a Python program using Matplotlib to plot a bar chart.
4. Write a Python program that uses list comprehension to create a new list based on an existing one.
5. Write a Python program that uses a lambda function and the filter() function to filter out even numbers from a list.
6. Write a Python program that uses a generator to generate the Fibonacci sequence.
7. Write a Python program that uses a decorator to time the execution of a function.
8. Write a Python program that uses regular expressions to validate an email address.
9. Write a Python program that uses the os module from the Python Standard Library to interact with the operating system.
10. Write a Python program that uses the datetime module from the Python Standard Library to work with dates and times.

Course Assessment (Full 50 Marks Internal Assessment)

Practical assessments are an effective way to gauge the skills and comprehension of programming concepts. Below are some practical assessment methods for a Python programming course:

1. Coding Assignments: The most common way to evaluate programming skills. These assignments can be small pieces of code related to the topic discussed in a particular unit. This allows students to apply the concepts they've learned.

2. Mini Projects: After completing a few units, students can be assigned mini projects that incorporate the topics covered. These projects should be larger than the usual coding assignments, giving the students an opportunity to combine different concepts they have learned.

3. Code Review: Pair up students and have them review and critique each other's code. They could check for errors, suggest better methods, or recommend different coding practices. This not only helps them learn to read others' code but also provides an understanding of best practices.

4. Debugging Exercises: Provide students with a piece of code that has bugs/errors in it. Their task would be to debug the code, correct it, and make it run successfully.

5. Whiteboard Coding: Although this is traditionally done in person, it can also be conducted virtually. Students are given a problem statement and they have to write code on the spot. It tests their problem-solving skills, ability to think under pressure, and their command over syntax.

6. Peer Programming: Pair students together to solve a problem. This can foster collaboration, improve problem-solving skills, and enhance code design skills.

7. Timed Challenges: This method adds an element of pressure. Students are given a time limit within which they have to solve a set of problems or tasks. This is a good way to prepare them for real-life situations, as often developers have to work under time constraints.

8. Coding Quizzes: These can be done online and are useful for quick revision and assessment of learning. They may consist of multiple-choice questions, fill in the blanks (code snippets), or output prediction questions.

9. Interactive Jupyter Notebook Assignments: Python Jupyter notebooks are great for this, as they allow you to write code and document it in one place. You can set tasks within the notebook that students need to complete.

10. Capstone Project: Towards the end of the course, a final, larger project that encompasses all the core areas of the curriculum can be assigned. This is typically a more complex problem statement or software development task.

Remember, the main goal is to ensure that students can not only write code but also think logically, troubleshoot issues, and understand the nuances of Python programming. These assessments will help enhance their problem-solving skills and confidence in programming.

Course Code: CS-441T(D)	Course Title:- Emotional Intelligence
Total Credit: 2	Marks: 50 (UA: 40 + IA: 10)
Periods: 3 per week (50 Minutes each)	
Prerequisites: There are no prerequisites for this course	
Learning Objectives	
<ol style="list-style-type: none"> 1. Understand the concept of Emotional Intelligence and its importance in personal and professional life 2. Develop skills for improving self-awareness, self-management, social awareness, and relationship management 3. Apply EI skills in personal and professional settings to enhance relationships, teamwork, and leadership 4. Create an action plan for continuous improvement of EI skills 	
Learning Outcomes	
<ol style="list-style-type: none"> 1. Improved self-awareness and ability to regulate emotions 2. Enhanced social awareness and empathy 3. Improved communication and relationship management skills 4. Increased leadership potential and effectiveness in the workplace. 	
Unit -I: Introduction to Emotional Intelligence (10 Periods)	
<ol style="list-style-type: none"> 1. What is Emotional Intelligence (EI)? 2. Why is EI important? 3. Understanding the four components of EI: self-awareness, self-management, social awareness, and relationship management. 	
Unit -II: Developing Self-Awareness and Self-Management Skills (10 Periods)	
<ol style="list-style-type: none"> 1. Assessing your EI using the Emotional Intelligence Appraisal 2. Strategies for improving self-awareness, including mindfulness and journaling 3. Techniques for improving self-management, including stress management, emotional regulation, and impulse control. 	
Unit -III: Developing Social Awareness and Relationship Management Skills (10 Periods)	
<ol style="list-style-type: none"> 1. Understanding social awareness and empathy 2. Developing relationship management skills, including communication, conflict resolution, and leadership 	

3. Building and maintaining positive relationships

Unit -IV: Applying EI in Personal and Professional Settings (10 Periods)

1. Applying EI in personal relationships, including family and friendships
2. Using EI in the workplace, including teamwork, leadership, and career development
3. Developing an action plan for improving EI skills and setting goals for personal and professional growth.

Unit-V: Test and Tutorials (05 Periods)

In addition to CIA, Tutorial, Seminars, Assignments & case studies are to be given for building proficiency in the course. (Respective course in-charge should maintain the records for the same).

Textbook

1. The Emotional Intelligence Handbook: A Complete Guide to Developing and Improving Your Emotional Intelligence by Anthony C. Mersino (2021)

References:

1. Emotional Intelligence: Why it can matter more than IQ by Daniel Goleman
2. Emotional Intelligence 2.0 by Travis Bradberry and Jean Greaves
3. The Emotional Life of Your Brain: How Its Unique Patterns Affect the Way You Think, Feel, and Live--and How You Can Change Them by Richard J. Davidson and Sharon Begley
4. Emotional Intelligence for Managers: Rise above the chaos of the workplace by R. Sridhar
5. The Power of Emotional Intelligence by Sanjay Singh
6. Emotional Intelligence at Work by Geetu Bharwaney
7. Mind Over Mood: Change How You Feel by Changing the Way You Think by Dennis Greenberger and Christine A. Padesky

E-Resources

2. Emotional Intelligence 2.0 website: <https://www.emotionalintelligence2-0.com/>
3. Greater Good Science Center at UC Berkeley: https://ggsc.berkeley.edu/topic/emotional_intelligence
4. MindTools Emotional Intelligence Toolkit: https://www.mindtools.com/pages/article/newCDV_59.htm
5. Harvard Business Review Emotional Intelligence articles: <https://hbr.org/topic/emotional-intelligence>
6. Psychology Today Emotional Intelligence articles: <https://www.psychologytoday.com/us/basics/emotional-intelligence>

Course Assessment (Full 50 Marks Internal Assessment)

Assessing the effectiveness of an emotional intelligence course can be challenging, but there are several practical assessments that can be used to demonstrate the skills acquired through the course. Here are some before and after tests that could be used to assess the effectiveness of an emotional intelligence course:

1. **Self-Assessment:** Before and after the course, students could be asked to complete a self-assessment of their emotional intelligence using a standardized tool such as the Emotional Intelligence Appraisal or the Mayer-Salovey-Caruso Emotional Intelligence Test. The results of these assessments could be compared to show any changes in their emotional intelligence.
2. **Role-Play Exercises:** Before and after the course, students could be asked to participate in a

role-play exercise that requires them to apply emotional intelligence skills. For example, they could be asked to role-play a difficult conversation with a coworker or friend, and the before and after conversations could be compared to demonstrate any improvements in their ability to manage their emotions and respond to others effectively.

3. **Feedback from Others:** Before and after the course, students could be asked to collect feedback from colleagues, friends, or family members on their emotional intelligence skills. They could ask for feedback on specific areas, such as empathy or self-awareness, and compare the feedback to demonstrate any improvements.
4. **Case Studies: Before and after the course,** students could be asked to analyze a case study that requires them to apply emotional intelligence skills. For example, they could be asked to analyze a workplace conflict and suggest solutions that demonstrate empathy and relationship management skills. The before and after case studies could be compared to show any improvements in their ability to apply emotional intelligence skills.
5. **Group Project:** Before and after the course, students could be asked to work on a group project that requires them to apply emotional intelligence skills. For example, they could be asked to plan a community service project or organize a charity event. The before and after projects could be compared to demonstrate any improvements in their ability to work collaboratively and manage relationships effectively.

These practical assessments can help demonstrate the effectiveness of an emotional intelligence course and show how students have acquired the necessary skills to apply emotional intelligence in their personal and professional lives.