Maharana Pratap Govt. Degree College Amb

District Una HP

Lesson Plans

by

Department of Computer Science

For

Bachelor of Science Physical Science (Physics, Computer Science and Mathematics) Annual Pattern

B.Sc. (Computer Science)

2022-23

www.mpgcamb.com | govtcollegeamb@gmail.com | mpgcamb@gmail.com |01976-26003

Table of Contents

Structure of Courses

Sr. No.	Course	Course Name	Course Code	Credits	Practical/Theory	Hours per week
1.	Core Course-II	Problem Solving Using Computer	COMP101TH	4	Theory	4
2.		Software Lab Using Python	COMP101PR	2	Practical	2
3.	Core Course-IV	Office Automation Tools	COMP102TH	4	Theory	4
4.		Office Automation Tools Lab	COMP102PR	2	Practical	2
5.	Core Course-VIII	Computer System Architecture	COMP201TH	6	Theory	6
6.	Core Course-XI	Database Management System	COMP202TH	4	Theory	4
7.		Database Management System Lab	COMP202PR	2	Practical	2
8.	SEC-2	PHP Programming	COMP203TH	4	Theory	4
9.	DSE-2A	Operating System	COMP301TH	6	Practical	6
10.	DSE-2B	Data Structure and File Processing	COMP302TH	4	Theory	4
11.		Data Structure and File Processing Lab	COMP302PR	2	Theory	2
12.	SEC 4	Software Engineering	COMP303TH	4	Theory	4
Total				**44**	**8-TH + 4 PR=12**	**44**

SEC-Skill Enhancement Elective Course, DSE-Discipline Specific Elective Course

Note: The above-mentioned subjects and credits are specifically for the subjects taught by Computer Science Department. Other subjects and credits i.e. Physics, Mathematics, English, and Environment are covered by the respective department.

Programme outcomes

After doing their B.Sc. the students are expected to go for admission to various post-graduate programmes like M.Sc. in Computer Science, Physics, and Mathematics. They can also go for post-graduate diploma courses like P.G.D.C.A etc. They can go for M.C.A., M.B.A. also. They can also compete in various competitive exams like HPAS, IAS, IPS, IRS, IFS, and Banking sector.

There are several courses that a student can pursue after completing a B.Sc. in Computer Science form Himachal Pradesh University, Shimla. More specifically, the most popular options for students after B.Sc. Computer Science are:

M.Sc. in Computer Science: This is a two-year postgraduate program that offers advanced knowledge and skills in computer science.

MCA (Master of Computer Applications): This is a three-year postgraduate program that focuses on computer applications and software development.

MBA in Information Systems: This is a two-year postgraduate program that offers knowledge and skills in managing information systems and technology in businesses.

PG Diploma in Data Science: This is a one-year postgraduate diploma that offers specialized knowledge in data analytics, machine learning, and other related areas.

PG Diploma in Cyber Security: This is a one-year postgraduate diploma that offers specialized knowledge in cyber security and related areas.

PG Diploma in Web Development: This is a one-year postgraduate diploma that offers specialized knowledge in web development and related areas.

M.Sc. in Artificial Intelligence: This is a two-year postgraduate program that offers specialized knowledge and skills in artificial intelligence, machine learning, and related areas.

M.Sc. in Information Technology: This is a two-year postgraduate program that offers advanced knowledge and skills in information technology.

However, there are many other options available as well, depending on the student's interests and career goals.

Programme Specific outcomes

1. The students gain the knowledge of systematic observations, model making, theoretical predictions thereby understanding various phenomena in nature.

2. To make the students think critically and to use appropriate concepts in problem solving.

3. To enhance the skills of the students to be innovative.

4. To provide the students skills to design and construct software applications and make them research oriented.

5. To make students sensitive to their surrounding and social issues, through projects.

6. To develop Ability in students to meet challenges and equip them to be competent.

7. To develop a qualitative and quantitative approach in the students.

8. To provide hands on experience to the students on sophisticated instruments and programming skills.

More specifically, this programme polishes many things of students including the following:

- To understand the fundamental concepts and theories of computer science, including programming, data structures, algorithms, computer organization, and operating systems.
- To develop proficiency in programming languages such as Python, HTML, and PHP.
- To gain knowledge of software engineering principles, including software design, testing, and documentation.
- To develop skills in database management, data mining, and data analysis.
- To develop skills in web development, including front-end and back-end development, and gain proficiency in web technologies such as HTML, CSS, PHP, and MySQL.
- To gain practical experience through hands-on projects and internships, and develop teamwork and collaboration skills.
- To develop communication skills and the ability to effectively communicate technical information to both technical and non-technical audiences.
- To gain knowledge of emerging trends and technologies in the field of computer science, and develop the ability to adapt and learn new technologies as needed.

Course Outcomes

Sr. No.	Year	Course Name	Course Code	Course Outcomes
1.	1st Year	Problem Solving Using Computer	COMP101TH	After completion of this course, students will have a strong understanding of the inner workings of computer systems, including the hardware and software components that make up a computer system. They will be able to understand the basic concepts of computer organization and architecture, such as memory, CPU, input/output devices, and storage. Furthermore, students will have a comprehensive understanding of the Python programming language, including its syntax, data structures, and control structures. They will learn how to write basic programs, work with files, and interact with the operating system using Python. Additionally, students will learn basic problem-solving techniques that are essential in the field of computer programming. This includes understanding the problem, identifying the requirements and constraints, breaking down the problem into smaller sub-problems, and developing an algorithmic solution using appropriate data structures and control structures. The course will also emphasize the significance of problem-solving techniques in the field of computer programming. Students will learn how to analyze and evaluate the efficiency and correctness of their programs, and develop strategies to improve their code. Overall, upon finishing this course, students will have a strong foundation in computer systems and programming concepts, and be able to apply their knowledge to real-world problems. They will have the ability to write basic programs in Python, understand the significance of problem-solving techniques, and have the skills necessary to evaluate and improve their code.
2.	1st Year	Software Lab Using Python	COMP101PR	Upon completing this course, students will have the necessary skills to design and develop basic programs using the Python programming language. They will be proficient in implementing essential programming constructs such as operators, data types, loops, functions, and objects, which are critical components in the development of efficient and effective software solutions. With this knowledge, students will be able to construct simple Python programs that perform a range of computational tasks, from basic arithmetic operations to more complex algorithms. Additionally, students will gain an understanding of the underlying principles and best practices that guide programming language usage, enabling them to develop high-quality, maintainable code that adheres to industry standards.
3.	1st Year	Office Automation Tools	COMP102TH	Upon completion of this course, students will possess a thorough understanding of various application tools available in MS Office and Open Office Libra. They will acquire the necessary skills to utilize essential tools such as MS Word, PowerPoint, and Excel, as well as their Open Office equivalents, Writer and Calc. Moreover, students will gain proficiency in creating compelling presentations using Impress Presentation and designing technical diagrams using Draw Drawing. Additionally, they will learn how to utilize Maths Formula to create and edit mathematical equations, and understand the fundamentals of Base Database for creating and managing databases. Overall, this course will provide students with an in-depth understanding of the capabilities and applications of MS Office and Open Office Libra tools, empowering them to utilize these tools effectively in various professional and academic settings.
4.	1st Year	Office Automation Tools Lab	COMP102PR	Upon completion of this course, students will have gained the proficiency to effectively use various essential tools such as MS Word, PowerPoint, and Excel, as well as their Open Office equivalents, Writer and Calc. Additionally, students will have acquired an understanding of advanced tools such as Impress Presentation, Draw Drawing, Math Formula, and others that are commonly utilized in office automation applications. They will have the knowledge necessary to design and create professional documents, presentations, spreadsheets, and technical diagrams using these tools. Furthermore, students will understand how to integrate the use of these tools in various office automation applications, such as data analysis, project management, and document collaboration. This course will enable students to use these tools to increase their productivity, streamline their workflow, and produce high-quality work in a timely manner.
5.	2nd Year	Computer System Architecture	COMP201TH	Upon completing this course, students will have a comprehensive understanding of the architecture and functioning of computer systems. They will be familiar with the critical components of computer system architecture, including gates, flip-flops, registers, counters, decoders, encoders, and data buses. Additionally, students will have an understanding of the complex workings of CISC and RISC computer architectures, as well as concepts such as DMA controller, addressing modes, stack organization, and more. Students

				will also acquire the necessary knowledge to comprehend the internal mechanisms of modern computer systems, including their various components such as the central processing unit (CPU), memory, input/output devices, and peripherals. They will have a thorough understanding of the interactions between these components, and how they work together to perform various computational tasks. Moreover, students will be able to apply this knowledge to practical situations, such as designing and building computer systems, troubleshooting system issues, and optimizing system performance. With this expertise, students will be well-equipped to pursue careers in various areas of computer science, including software engineering, system administration, and computer hardware design.
6.	2nd Year	Database Management System	COMP202TH	After completion of this course, students will have a comprehensive understanding of the basics of Database Management System (DBMS). This includes the principles and concepts of DBMS, the architecture and design of database systems, and the various applications of DBMS. Some of the critical components of DBMS that the students will learn are normalization, Relational Database Management System (RDBMS) and MySQL. To gain practical experience, students will also learn about the technicalities of Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language (DCL). DDL is used to define the structure of a database, DML is used to manipulate data in the database, and DCL is used to control access to the database. Overall, upon completion of this course, students will have a solid understanding of DBMS and its critical components, and be able to apply this knowledge to real-world scenarios.
7.	2nd Year	Database Management System Lab	COMP202PR	Upon completion of the MySQL component of the course, students should be able to understand many things including the basic principles and concepts of MySQL as a Relational Database Management System (RDBMS), Use MySQL to create, modify and delete databases, tables, and indexes, Understand and utilize different data types in MySQL, Write queries using SQL to retrieve, filter, and sort data from MySQL databases, Understand and apply different SQL functions in MySQL, Utilize subqueries and joins in, MySQL to combine and analyze data from multiple tables, Manage data in MySQL databases using Data Manipulation Language (DML) commands such as INSERT, UPDATE, DELETE, Understand the importance of data security and use Data Control Language (DCL), commands to grant and revoke privileges to MySQL users, Design and implement MySQL databases to meet specific business requirements, Optimize MySQL databases for better performance and scalability by understanding and utilizing indexing, partitioning, and query optimization techniques.
8.	2nd Year	PHP Programming	COMP203TH	Upon completion of the PHP course, students should be able to understand the basic principles and concepts of PHP as a server-side programming language, Write PHP code to dynamically generate HTML content for web pages, Utilize different data types in PHP and understand how to manipulate them, Understand and use control structures such as if-else statements, loops, and switch statements in PHP, Work with arrays and understand their use cases in PHP programming, Utilize functions and classes in PHP to create reusable code, Understand and work with different types of files in PHP, including reading, writing, and manipulating file content, Utilize PHP to interact with databases and perform CRUD operations (Create, Read, Update, Delete), Utilize PHP to develop web applications with forms, user authentication, and session management.
9.	3rd Year	Operating System	COMP301TH	Upon completion of the Operating System course, students should be able to understand the basic principles and concepts of operating systems, including the role of the operating system, process management, memory management, and file systems. Describe the architecture and components of different operating systems, including Windows, Unix, and Linux. Understand the different types of processes, threads, and synchronization mechanisms used in operating systems. Utilize different algorithms for CPU scheduling, process synchronization, and deadlock avoidance in operating systems. Understand the different memory management schemes, including virtual memory and paging. Understand the role of file systems and how they manage files and directories on disk. Utilize command-line interfaces in different operating systems to navigate the file system and perform basic tasks. Understand the importance of security in operating systems and how it is implemented through access control, authentication, and encryption. Understand the concept of virtualization and its role in operating system management and resource allocation. Utilize operating system tools and utilities for system monitoring, troubleshooting, and performance tuning.
10.	3rd Year	Data Structure and File Processing	COMP302TH	Upon completion of the Data Structure and File Processing course, students should be able to understand the basic principles and concepts of data structures, including arrays, linked lists, stacks, queues, trees, and graphs.

				Describe the characteristics and use cases of different data structures, including their efficiency in terms of time and space complexity. Implement various data structures in programming languages C++. Analyze the performance of data structures in terms of time and space complexity. Understand the concept of algorithm analysis and apply it to evaluate the performance of algorithms and data structures. Understand the concept of file processing and the different types of file organization methods, including sequential, indexed sequential, and direct. Understand the concept of hashing and its use in efficient search and retrieval of data. Implement file processing and algorithms in programming languages C++. Utilize data structures and file processing to solve real-world problems, such as sorting and searching large amounts of data. Understand the importance of data and file structure design in the development of efficient and scalable software applications.
11.	3rd Year	Data Structure and File Processing Lab	COMP302PR	Upon completion of the Data Structure and File Processing Lab course, students should be able to demonstrate proficiency in implementing data structures, such as arrays, linked lists, stacks, queues, trees, and graphs in C++ programming language. Design and implement algorithms using data structures to solve a variety of problems, such as sorting and searching data. Understand the concept of file processing and implement file processing techniques using C++ programming language. Demonstrate an understanding of the different types of file organization methods, including sequential, indexed sequential, and direct, and implement them using C++ programming language. Analyze the performance of algorithms and data structures in terms of time and space complexity and identify areas for improvement. Demonstrate an understanding of the importance of data and file structure design in the development of efficient and scalable software applications. Utilize C++ programming language to create efficient and scalable software applications that implement data structures and file processing techniques. Understand the importance of testing and debugging in software development, and apply these skills to identify and fix errors in their code. Work collaboratively in a team environment, sharing code and ideas, and communicating effectively to achieve common goals. Develop problem-solving skills, and be able to approach a new problem, identify the data structures and algorithms that may be suitable for solving it, and implement a solution using C++ programming language.
12.	3rd Year	Software Engineering	COMP303TH	Upon completion of the Software Engineering course, students should be able to understand the fundamental concepts of software engineering, including software development lifecycle, software requirements, software design, testing, and maintenance. Utilize various software development methodologies, such as Waterfall, Agile, and DevOps, to develop software applications. Understand the importance of software requirements, and be able to elicit, analyze, specify, and validate requirements using appropriate tools and techniques. Design software applications that meet the requirements and specifications, using various design principles, such as object-oriented design, modular design, and architectural design. Understand the importance of software testing, and be able to develop and implement testing strategies, including unit testing, integration testing, and acceptance testing. Understand the principles of software maintenance, and be able to apply appropriate techniques to maintain and update software applications. Work collaboratively in a team environment, using appropriate tools and techniques for project management, communication, and collaboration.

Problem Solving Using Computer COMP101TH

Course duration: 16 weeks

Unit-I: Computer Fundamentals

Week 1: Introduction to the course and syllabus, Introduction to Computers, Definition and characteristics of computers, Uses of computers, Types, and generations of computers.

Week 2: Basic Computer Organization, Units of a computer CPU, ALU.

Week 3: Memory hierarchy, registers, and I/O devices.

Unit-II: Planning the Computer Program

Week 4: Concept of Problem Solving, Problem definition, Program design, Debugging

Week 5: Types of Errors in Programming, Syntax errors, Runtime errors, Logical errors

Week 6: Documentation, Importance of documentation, Types of documentation

Week 7: Techniques of Problem Solving, Flowcharting, Decision table, Algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming

Unit-III: Overview of Programming with Python

Week 8: Structure of a Python Program, Elements of Python

Week 9: Introduction to Python, Python Interpreter, Using Python as a calculator, Python shell, Indentation

Week 10: Atoms, Identifiers, and Keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bitwise operator, Increment or Decrement operator)

Week 11: Creating Python Programs, Input and Output Statements, Control statements (Looping - while Loop, for Loop, Loop Control, Conditional Statement - if...else, Difference between break, continue and pass)

Unit-IV: Structures and Introduction to Advanced Python

Week 11: Structures, Numbers, Strings, Lists, Tuples, Dictionary.

Week 12: Date & Time, Modules, Defining Functions, Exit function, Default arguments

Week 13: Introduction to Advanced Python, Objects and Classes.

Week 14: Inheritance, Regular Expressions.

Week 15: Event Driven Programming.

Week 16: GUI Programming

Practical Learning Environment (if Required):

Python programming environment “Thonny”

Note: This lecture plan is tentative and subject to change depending on the pace of learning and other factors.

Office Automation Tools COMP102TH

Course duration: 16 weeks

Unit I: Introduction to open office/MS office/Libra office

Week 1: Introduction to the course and syllabus, overview of office automation tools: open office/MS office/Libre office

Week 2: Differences between the various office automation tools, Basic features and tools common to all office automation tools

Unit II: Word Processing

Week 3: Word processing software: Introduction and overview, Creating and formatting documents in word processing software, formatting text: font, size, style, color, etc.

Week 4: Creating and formatting pages: margins, orientation, page breaks, etc.

Week 5: Creating and formatting lists: bullets, numbering, indentation, etc., Creating and formatting tables: table layout, cell formatting, table styles, etc.

Week 6: Advanced formatting techniques: headers, footers, footnotes, endnotes, etc., Creating and formatting templates for consistency and efficiency

Unit III: Spreadsheet

Week 7: Introduction to spreadsheet software: Overview and basic features, Creating and formatting worksheets: rows, columns, cells, cell formatting, etc.

Week 8: Working with data: sorting, filtering, grouping, subtotalling, etc., Creating and formatting charts and graphs

Week 9: Using formulas and functions: basic arithmetic, statistical, logical, and lookup functions, Creating and using macros for automating repetitive tasks

Week 10: Introduction to Pivot Table for analyzing and summarizing data

Unit IV: Presentation Tools

Week 11: Introduction to presentation software: Overview and basic features, Creating and formatting slides: slide layout, slide backgrounds, adding text and images, etc.

Week 12: Formatting text: font, size, style, color, etc., Creating and formatting objects: charts, diagrams, shapes, etc.

Week 13: Advanced formatting techniques: adding animations, transitions, multimedia elements, etc.

Week 14: Creating and formatting notes and handouts for presentations, Creating and running slide shows, Tips for effective and engaging presentations

Week 15: Recap of all topics covered, Review of the tools and techniques for efficient and effective use of office automation tools

Week 16: Strategies for continuous learning and keeping up with updates and new features, Final project or assignment, applying the skills and knowledge learned in the course

Practical Learning Environment (if Required):

Open Office, MS Office, and Libra Office.

Note: This lecture plan is tentative and subject to change depending on the pace of learning and other factors.

Computer System Architecture **COMP201TH**

Course duration: 16 weeks

Unit I: Introduction, Data Representation, and basic computer Arithmetic

Week 1: Introduction to computer system architecture, Logic gates, and Boolean algebra

Week 2: Combinational circuits and circuit simplification, Flip-flops, and sequential circuits

Week 3: Decoders and multiplexors, Registers and counters, Memory units and their types, Number systems and their conversions

Week 4: Complements and their types, Fixed and floating point representation of numbers, Character representation and ASCII code, Addition, and subtraction of numbers

Unit II: Basic Computer Organization and Design, and a central processing unit

Week 5: Magnitude comparison of numbers, Basic computer organization, and design

Week 6: Computer registers and bus system, Instruction set, and timing and control

Week 7: Instruction cycle and memory reference, Input-output and interrupt

Week 8: Central Processing Unit (CPU) and its organization, Arithmetic and logical micro-operations

Week 9: Stack organization and micro-programmed control

Unit III: Programming the Basic Computer

Week 10: Instruction formats and addressing modes

Week 11: Instruction codes and machine language

Week 12: Assembly language and input-output programming

Unit IV: Input-output organization

Week 13: Peripheral devices, I/O interface, and modes of data transfer

Week 14: Direct memory access (DMA), and its types

Week 15: Review of all topics covered

Week 16: Question and answer session, practice problems and exercises

Practical Learning Environment (if Required):

Not Applicable

Note: This lecture plan is tentative and subject to change depending on the pace of learning and other factors.

Computer Database Management System COMP202TH

Course duration: 16 weeks

Unit I: Introduction to Database Management Systems

Week 1: Introduction to Database Management Systems, Characteristics of the database approach

Week 2: Data models

Week 3: DBMS architecture, and data independence.

Unit II: Entity-Relationship Modeling and Enhanced ER Modeling

Week 4: Entity-Relationship Modeling: Entity types, relationships, attributes, keys, and participation constraints.

Week 5: Enhanced Entity-Relationship Modeling: Subtypes, supertypes, specialization, and generalization.

Week 6: SQL-99, Schema Definition, CREATE TABLE, ALTER TABLE, DROP TABLE.

Week 7: Constraints, and data types.

Unit III: Relational Data Model

Week 8: Basic concepts of relational data model, relational constraints.

Week 9: Relational algebra.

Week 10: SQL queries.

Unit IV: Database Design

Week 11: ER and EER to Relational Mapping.

Week 12: Functional Dependencies and Normalization, First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF).

Week 13: Fourth Normal Form (4NF), Fifth Normal Form (5NF).

Week 14: Review of all the topics under *Unit I*, and *Unit II*.

Week 15: Review of all the topics under *Unit III*, and *Unit IV*.

Week 16: Question and answer session, practice problems and exercises.

Practical Learning Environment (if Required):

MySQL

Note: This lecture plan is tentative and subject to change depending on the pace of learning and other factors.

PHP Programming **COMP203TH**

Course duration: 16 weeks

Unit I: Introduction to PHP

Week 1: Introduction to PHP, Software requirements and tools for PHP development

Week 2: Basic syntax of PHP, PHP variables and constants

Week 3: Types of data in PHP, Expressions in PHP

Unit II: Handling HTML forms with PHP, PHP Conditional events and Loops

Week 4: Handling HTML forms with PHP, Capturing form data using GET and POST methods

Week 5: Dealing with multi-value fields, redirecting a form after submission

Week 6: Conditional statements in PHP: IF-ELSE, nested IF-ELSE, Switch case statements in PHP

Week 7: Loops in PHP: while, for, do-while, Control statements in PHP: Goto, Break, Continue, Exit

Unit III: PHP Functions and Array

Week 8: PHP Functions: Declaration, calling, and need for functions

Week 9: PHP Functions with arguments: Default arguments, Call by Value and Call by Reference, and Global and Local scope of a function

Week 10: Anatomy of an Array, Creating index-based and associative arrays, and accessing arrays in PHP

Week 11: Looping through index-based and associative arrays using each() and foreach(), Useful Library Functions for arrays

Unit IV: String manipulation and regular expression

Week 12: Creating and accessing strings in PHP, Searching and replacing strings in PHP

Week 13: Formatting, joining, and splitting strings, and Library Functions related to strings in PHP

Week 14: Introduction to Regular Expression in PHP, Use and advantage of Regular Expression over inbuilt functions

Week 15: Using preg_match(), preg_replace(), and preg_split() functions for Regular Expression in PHP

Week 16: Review all topics covered in the previous weeks, and Practicing exercises and examples for each topic.

Practical Learning Environment (if Required):

WAMP Server and MySQL

Note: This lecture plan is tentative and subject to change depending on the pace of learning and other factors.

Operating System **COMP301TH**

Course duration: 16 weeks

Unit I: Introduction and Types of Operating Systems

Week 1: Introduction to operating system: system software, resource abstraction, OS strategies

Week 2: Types of operating systems: Multiprogramming, Batch, Time Sharing

Week 3: Types of operating systems: Single user and Multiuser, Process Control & Real-Time Systems

Unit II: Operating System Organization, Process Management, and Scheduling

Week 4: Operating System Organization: Factors in operating system design, basic OS functions, implementation consideration

Week 5: Process modes, methods of requesting system services – system calls and system programs

Week 6: Process Management: System view of the process and resources, initiating the OS, process address space

Week 7: Process Management: Process abstraction, resource abstraction, process hierarchy, Thread model

Week 8: Scheduling Mechanisms, Strategy selection, non-pre-emptive and pre-emptive strategies

Unit III: Memory Management

Week 9: Memory Management: Mapping address space to memory space, memory allocation strategies

Week 10: Memory Management: Fixed partition, variable partition, paging, virtual memory

Unit IV: Shell Introduction and Shell Scripting

Week 11: Introduction to Shell: Shell and various types of shell, and various editors present in Linux, Different modes of operation in vi editor

Week 12: Shell Scripting: What is shell script, Writing and executing the shell script, and Shell variable (user defined and system variables)

Week 13: System calls: Using system calls, Advanced Shell Scripting: Loops, Branching, File Operations

Week 14: Advanced System Administration: Managing Users, Managing Services, Basic Networking Concepts

Week 15: Review all the topics under *Units I to IV*.

Week 16: Question and answer session, practice problems, and exercises.

Practical Learning Environment (if Required):

Unix Red Heat, Linux, Ubuntu, or Open SuSE, etc.

Note: This lecture plan is tentative and subject to change depending on the pace of learning and other factors.

Data Structure and File Processing COMP302TH

Course duration: 16 weeks

Unit I: Basic Data Structures

Week 1: Introduction to Data Structures, Abstract Data Structures: Stacks, Queues, Basic Operations on Stacks and Queues

Week 2: Linked Lists: Introduction and Implementation, Operations on Linked Lists

Week 3: Binary Trees: Introduction and Implementation, and Balanced Trees: AVL Trees and Red-Black Trees

Unit II: Searching

Week 4: Searching Algorithms: Linear Search, Binary Search, Internal and External Searching

Week 5: Memory Management: Garbage Collection Algorithms

Week 6: Storage Allocation for Objects with Mixed Size

Unit III: Physical Devices

Week 7: Characteristics of Storage Devices: Disks and Tapes, and I/O Buffering

Week 8: Basic File System Operations: Create, Open, Close.

Week 9: Extend, Delete, Read-block, Write-block, Protection Mechanisms

Unit IV: File Organization

Week 10: Sequential File Organization: Introduction and Implementation

Week 11: Indexed Sequential File Organization: Introduction and Implementation

Week 12: Direct and Inverted File Organization, and Multi-List and Directory Systems

Week 13: Indexing using B-Tree and B+ Tree

Week 14: Introduction to File Processing, File Processing in PHP: Reading and Writing to Files, File Pointers, and Examples of File Processing Applications in PHP.

Week 15: Review all the topics under *Units I to IV*.

Week 16: Question and answer session, practice problems, and exercises.

Practical Learning Environment (if Required):

C++

Note: This lecture plan is tentative and subject to change depending on the pace of learning and other factors.

Software Engineering **COMP303TH**

Course duration: 16 weeks

Unit I: Software Process, Software Requirements

Week 1: Introduction to Software Engineering: Definition, Importance, and Paradigms, Software Life Cycle Models: Waterfall, Incremental, Spiral, and Agile

Week 2: System Engineering and Development Process, Software Verification, Validation, and Testing

Week 3: Software Requirements: Functional and Non-functional, User and System, Requirements Elicitation, Validation, and Management

Week 4: Software Prototyping and Rapid Prototyping Techniques, Analysis and Modeling: Data, Functional, and Behavioral Models

Unit II: Design Concepts and Principles

Week 5: Modular Design, Design Heuristics, and Design Model and Document

Week 6: Architectural Design: Software Architecture, Data Design, and User Interface Design

Week 7: Real-Time Systems and Software Design

Unit III: Software Configuration Management and Software Project Management

Week 8: Software Configuration Management: The SCM Process, Version Control

Week 9: Change Control, Configuration Audit, and SCM Standards

Week 10: Software Project Management: Measures and Measurements, Estimations, and Empirical Estimation Models, and Project Scheduling

Unit IV: Testing, Trends in Software Engineering

Week 11: Software Testing: Taxonomy, Levels, Test Activities, and Types of Software Test, Black Box Testing, and Boundary Conditions

Week 12: Structural Testing and Test Coverage Criteria Based on Data Flow, Regression Testing, Testing in the Large, and Debugging

Week 13: Software Testing Strategies: Strategic Approach and Issues, Unit Testing, Integration Testing, Validation Testing, System Testing, and Debugging

Week 14: Trends in Software Engineering: Reverse Engineering and Re-engineering, Wrappers, and CASE Tools

Week 15: Introduction to Software Quality Assurance: Definition, Importance, and Principles, Quality Assurance Processes and Techniques, Software Reliability and Dependability, Software Maintenance

Week 16: Software Metrics and Measurements: Definition and Importance, Types of Metrics and Measurements. Software Engineering Ethics and Professional Practices, Emerging Trends and Future Directions in Software Engineering.

Practical Learning Environment (if Required)
Not Applicable
Note: This lecture plan is tentative and subject to change depending on the pace of learning and other factors.

Principal
M. P. Govt. College, Amb
Distt. Una (H.P.)