4TH SEMESTER BSc. CS(H)			
	SUB CODE		SUB NAME
MAJOR	CORE-I	PAPER-8	Web Technologies
	CORE-I	PAPER-9	OOP Using C++
	CORE-I	PAPER-10	Computer Network
MINOR	CORE-III	PAPER-2	Mechanics
	INTERNSHIP		

Core VIII

Web Technologies

Course Objectives:

- To understand the essentials of Web Technologies.
- To understand frontend (HTML, CSS, JavaScript) and backend Technologies (PHP) for developing Web Sites.

Learning Outcomes:
Upon completion of this course, students will be able to:

1. Learn the basics of Internet protocols and HTML
2. Learn the use of CSS
3. Learn the use of Java Scripts
4. Learn the use of PHP and design a Website

Unit-I:

Introduction to the Internet, Internet Protocols, World Wide Web (WWW): Introduction, History, HTTP and HTTP methods, Web Browser, Web Server and their examples, Web page, working principles of WWW. Web Development: Introduction, Front-end and Backend Development Technologies. Concepts of Client-Server communication.
HTML: Introduction, characteristics, basic structure of an HTML document, understanding basic HTML tags and attributes, creating an HTML document. Working with tags for text-formatting, lists, hyperlink, images, tables, frames, multimedia. HTML forms and its basic elements.

Unit-II:
Cascading Style Sheets (CSS): Introduction, Benefits of using CSS, Understanding the Syntax, CSS Selectors, Using CSS: External, Internal Inline CSS. Comments in CSS. Basic CSS Properties: Color, Background, Text, Font, List, Table, Display.
CSS Box Model: Introduction, working with Margin, Border, and Padding. Pseudo-class & element, working with block elements, Scrolling text, Navigation Bar and Drop Downs.

Unit-III:

JavaScripts: Introduction, Features, Benefits, Creating Simple JavaScript. Using JavaScript in HTML. Exploring Popup Boxes: alert, confirm, prompt box. Displaying outputs in JavaScript. Programming using JavaScript: Data types, Variables, Operators, Expressions (Arithmetic, String, Logical), Comments. Control Statements: Conditional, Looping and Jump Statements. Functions (built-in & user defined) and their usage. Working with Array and Date Objects. Introduction to DOM, Event handling and Form validation in JavaScript.

Unit-IV:

PHP: Features, Print/echo statement, Data Types, Variables, Constants, Strings, Arrays, Operators. Control Structures: Conditional, Looping & Jump Statements. Brief overview of Arrays, Functions: String, Date-Time, Mathematical and User-defined functions.
Embedding PHP in HTML, Reading Form data of a Web Page. Introduction to PHP with Database: Connecting to Database, Creating Table, inserting records, modifying data and retrieving data and displaying in HTML.

Text Book:

Web Technologies (Black Book), DreamTech Press

Reference Books:

- ✓ *Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP 4th Edition by Ivan Bayross.*
- ✓ *HTML, XHTML and CSS Bible, 5ed, Willey India-Steven M. Schafer.*

Core VIII- Lab: Web Technologies

1. Create a Web Page to display "Hello HTML".
   - Display the same using different headings: h1 to h6
   - Apply bold, italic formats
   - Change text color, background of the page

2. Create a Web Page to display the list of Fruits using both ordered and unordered list. Recreate the lists using images of the same items.
3. Use the web page created in Question Number 2. When the user clicks on the image of a Fruit, it should open a new page that contains an image of the Fruit along with its benefits (use multiple paragraphs, and make sure the image is aligned properly).
4. Create a web page that displays details of the Fruits in Tabular format. Use serial no., name, color, taste, price/kg.
5. Create a Web Page that displays a video file (record/create a video of your own).
   - Apply various controls such as paly, pause, volume.
   - Apply autoplay, muted and both.
   - Display a Youtube Video in your Web Page.

6. Design a Student registration form to collect various data about a Student which includes Name, Age, Gender (M, F, O), Mobile No., Email ID, Stream (Science/Arts/Commerce in drop-drown), Choice for participating in NCC, NSS, YRC (use check box), and two buttons for reset and submit respectively. Display the form at the center of the page with proper alignment of each item in the form.
7. Use the web page created in Question no. 4. Use CSS and apply various styling to the text, colors to each row of the table, styling to borders and background color of the table.
8. Create a Web page with a Horizontal Navigation bar containing four items such as Home, College, Students, Teachers. The first item should be active, by default. The background color of the item changes when the user moves the mouse over it.
9. Create a Web Page that asks the user to enter the number of Students, and then iteratively ask the details of each Student and display them in a list, using JavaScript.
10. Modify the above program (Q. 9) to display the details in a tabular format, dynamically using JavaScript.
11. Create a Web Page with two text fields and four buttons to perform arithmetic operations such as Addition, Subtraction, Multiplication and Division. The user has to enter numbers in the text fields and press any of the above buttons. The JavaScript progrm shoud perform corresponding operations and display the result in the same page. [NOTE: Use different functions for each operation]
12. Use JavaScript and validate the form data in the Student registration Page (created in Q. no. 6). When the user clicks the Submit button, the program has to validate that every field contains valid data. [NOTE: validate name, age, gender, email, mobile number]
13. Write a PHP program that asks the user to enter a number and finds the factorial of it.
14. Write a PHP program that creates a Table in a database with a number of columns as determined by the fields in the Student registration form created above.
15. Write a PHP program to store the data of the Student registration form in a Database.
16. Write a PHP program that asks the user to enter a Student name and display the Student details retrieved from the database in the same page.
17. Students are required to combine the Student Registration form, and PHP program (s) [which interact with Database] to see the dynamic updation of the Student registration data in the Database when a new Student is registered.

Core-IX **Object Oriented Programming using C++**

Course Objectives:

- To know about the Object-Oriented Programming concepts.
- To write object-oriented programs using C++ constructs

Learning Outcomes:
Upon completion of this course, students will be able to:

- Understand OOPs concepts as a programming style
- Use class/objects in programs and functions of different types
- Learn the concept of inheritance and overloading of functions and operators
- Use files in C++

Unit-I:

Principles of Object-Oriented Programming: Object-Oriented Programming (OOP) Paradigm, Basic Concepts of OOP, Benefits of OOP, Characteristics of OOPS, ObjectOriented Languages, Applications of OOP.
Introduction to C++, Difference between C & C++, Tokens, Data types, Operators, Structure of C++ Program, C++ statements, Expressions and Control Structures.
Functions in C++: Argument passing in function, Inline Functions, DefaultArguments, Const. Arguments, Friend function.

Unit-II:

Classes and Objects: Defining Member Functions, Making an outside Function Inline, Nested Member Functions, Private Member Functions, Arrays within a Class, Memory Allocation for Objects, Static Data Members, Static Member Functions, Arrays of Objects, Objects as Function Arguments, Friend Functions.
Constructors & Destructors: Constructors, Parameterized Constructors, Constructors with Default Arguments, Dynamic Initialization of Objects, Copy Constructor, Dynamic Constructors, Destructors.

Unit-III:

Inheritance: Basics of Inheritance, Type of Inheritance, Virtual Base Classes, Abstract Classes, Member Classes, Nesting of Classes. Polymorphism: Pointers, Pointers to Objects, this Pointer, Pointers to Derived Classes, Virtual Functions, Pure Virtual Functions, Function Overloading, Operator Overloading.

Unit-IV:

Managing Console I/O Operations: C++ Streams, C++ Stream Classes, Unformatted I/O Operations, Formatted Console I/O Operations, Managing Output with Manipulators.

Files: Classes for File Stream Operations, Opening and Closing a File, Detecting end-of-file, File Modes, File Pointers and their Manipulations, Sequential Input and OutputOperations, Updating a File: Random Access, Error Handling during File Operations, Command-line Arguments.

Text Books:

- ✓ *E. Balgurusawmy, Object Oriented Programming with C++, 4/e (TMH).*
- ✓ *Paul Deitel, Harvey Deitel, "C++: How to Program", 9/e. Prentice Hall.*

Reference Books:

- ✓ *Bjarne Stroustroup, Programming - Principles and Practice using C++, 2/e, Addison-Wesley*
- ✓ *Herbtz Schildt, C++: The Complete reference, McGrawHill.*

Core IX- Lab: Programming using C++

1. Write a Program for Swapping of two numbers.
2. Write a Program to find sum of four numbers using default argument passing.
3. Write a Program to find square and cube of a number using inline function.
4. Write a Program to find the factorial of a number.
5. Write a Program to find reverse of a number.
6. Write a program to find sum of four numbers using default argument passing inmember function.
7. Write a Program to find area of circle, triangle and rectangle using functionoverloading.
8. Write a program to distinguish the properties of static and non-static ata members.
9. Write a program to show the method of accessing static private member function.
10. Write a program to show the ways of calling constructors and destructors.
11. Write a program to perform ++ operator overloading using member function.
12. Write a program to perform ++ operator overloading using friend function.
13. Write a program to perform + operator overloading for two complex number addition.
14. Write a program to perform + operator overloading for string concatenation.
15. Write a program to perform single inheritance.
16. Write a program to perform multiple inheritance.
17. Write a program to create an integer array using new operator and find the sum andaverage of array elements.
18. Write a program to implement virtual destructor.

19. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
20. Write a program to Copy the contents of one file to other.

Core-X **Computer Network**

Course Objectives:

- To understand data communication and network concepts.
- To learn about different communication standards
- To understand different network protocols

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand concepts on data communication and the use of communication devices
- Learn about analog and digital signals and basic components of data communication
- Learn about errors during data communication & access control mechanisms
- Learn various network protocols and network security issues

Unit-I:

Introduction to Data Communications and Network Models: Protocols and Standards, Layers in OSI Models, Analog and Digital Signals, Transmission Modes, Transmission Impairment, Data Rate Limits, Performance, Digital Transmission, Network Devices & Drivers: Router, Modem, Repeater, Hub, Switch, Bridge (fundamental concepts only).

Unit-II:

Signal Conversion: Digital-to-Digital Conversion, Analog-to-Digital Conversion, Digital-to-analog Conversion, Analog-to-Analog Conversion. Transmission Media: Guided Media, Unguided Media, Switching Techniques: Packet Switching, Circuit Switching, Datagram Networks, Virtual-Circuit Networks, and Structure of a Switch.

Unit-III:

Error Detection and Correction: Checksum, CRC, Data Link Control: Framing, Flow and Error Control, Noiseless Channels, Noisy channels, (Stop and Wait ARQ, Sliding Window Protocol, Go Back N, Selective Repeat) HDLC, Point-to-Point Protocol. Access Control: TDM, CSMA/CD, and Channelization (FDMA, TDMA, and CDMA).

Unit-IV:

Network Layer: Logical Addressing, IPv4 Addresses, IPv6 Addresses, Virtual-Circuit Networks: Frame Relay and ATM, Transport Layer: Process-Process Delivery: UDP, TCP. Application layers: DNS, SMTP, POP, FTP, HTTP, Basics of WiFi (Fundamental concepts only), Network Security: Authentication, Basics of Public Key and Private Key, Digital Signatures and Certificates (Fundamental concepts only).

Text Book:

- ✓ *Data Communications and Networking, Fourth Edition by Behrouza A. Forouzan, TMH.*

Reference Book:

- ✓ *Computer Networks, A. S. Tanenbaum, 4th edition, Pearson Education*.

Core X- Lab: Computer Network

1. Use the ***ipconfig*** (Windows) or ***ifconfig*** (Linux/Mac) command to display the current network configuration.
    i. Identify and document the IP address, subnet mask, and default gateway of the system.
    ii. Change the IP address of the system using ***netsh*** (Windows) or ***ifconfig*** (Linux/Mac). Verify the change using the same command.
    iii. Experiment by configuring static IP, dynamic IP.
2. Use the ***ping*** command
    i. to check connectivity between Systems in your Lab.
    ii. to a remote server (e.g., google.com).
    iii. Analyze the round-trip time and packet loss.
3. Use the ***tracert*** (Windows) or ***traceroute*** (Linux/Mac) command to trace the path to a remote server. Document the intermediate hops and their IP addresses.
4. Use the ***netstat*** command to display active connections, listening ports, and network statistics.
    i. Document and explain the various parameters and their significance.
    ii. Use ***netstat -r*** or ***route*** to display the routing table of your system. Identify the default gateway and other routes.
5. Use the ***arp -a*** command to display the ARP table of your system.
    i. Identify the MAC addresses corresponding to different IP addresses.
    ii. Clear the ARP cache using ***arp -d*** and verify the cache is cleared. Re-populate the ARP table by pinging different hosts on the network and verify the entries.
6. Use the ***nslookup*** command to query the DNS records of a domain (e.g., google.com).
    i. Identify and document the IP addresses associated with the domain.
    ii. Use the ***dig*** command (Linux/Mac) for a more detailed DNS query and compare the output with ***nslookup***
7. Use the ***nmcli*** command (Linux) or ***netsh wlan show networks*** (On Windows) to scan for available Wi-Fi networks and connect to a specified network. Document the steps and verify the connection.
8. Use the ***tcpdump*** command (Linux) or ***Wireshark*** to capture network packets.
    i. Capture and analyze traffic for a specific protocol (e.g., HTTP) and identify key details like source and destination IPs, ports, and packet content.
    ii. Filter captured packets to display only traffic to/from a specific IP address or port using ***tcpdump*** for Wireshark filters.
9. Use the ***nmap*** command to perform a network scan of your local network.
    i. Identify active hosts, open ports, and running services.
    ii. Perform a more detailed scan with service/version detection using ***nmap -sV*** and analyze the results.
10. Use the ***iptables*** command (Linux) to set up basic firewall rules. On Windows, use ***netsh advfirewall.*** Block all incoming traffic except for SSH and HTTP, and verify the rules are working.
11. Use the ***route*** command to add a static route to a specific network.
    i. Verify the route using ***route -n*** (Linux) or ***route print*** (Windows).

ii. Set up IP forwarding on a Linux system using ***sysctl*** to enable packet forwarding. Test the configuration by pinging through the system acting as a router.