DEPARTMENT OF COMPUTER SCIENCE
KUMAUN UNIVERSITY, NAINITAL

DRAFT SYLLABUS

National Education Policy-2020

Common Minimum Syllabus for Uttarakhand State Universities and Colleges

Four Year Undergraduate Programme- FYUP/Honours Programme/Master in Science

SUBJECT: COMPUTER SCIENCE

EFFECTIVE FROM ACADEMIC YEAR 2025-2026

Expert Committee

Prof. Ashish Mehta, Head and Convener, Department of Computer Science.
Kumaun University, Nainital.

Dr. Parul Saxena, Head, Department of Computer Science and IT.
SSJU Campus, Almora

Prof. M.P. Thapalyal, Department of Computer Science,
HNB Garhwal University

Dr. Shikha Jain, Department of Computer Science,
Delhi University

Syllabus Drafting Committee

Convenor Syllabus Drafting Committee Department of Computer Science, Kumaun University, Nainital	Prof. Ashish Mehta, Head and Convener
Kumaun University, Nainital	Prof. Ashish Mehta Mr. Anand Kumar Mr. Ashish Bhatt Mr. Hem Chandra Bhatt Mr. Mohd. Rehan Mr. Harshvardhan Chaudhary
Department of Computer Science, SSJ University, Almora	Dr. Parul Saxena, Head and Convenor

Drafting Assistance: Arpita Joshi and Neelam Anand. Both are Research Students of the Department of Computer Science.

Abbreviations & Acronyms	
DSC	Discipline Specific Course
DSE	Discipline Specific Electives
IAPC	Internship/Apprenticeship / Project/ Community Outreach
GE	Generic Elective
AEC	Ability Enhancement Course
IL	Pool of Indian Languages in the 8th schedule of the Constitution
L | T | P	Lecture | Tutorial | Practical

Semester Wise DSC/DSE/GE/SEC Courses List

Semester	DSC	DSE	GE	SEC
I	Computer Fundamentals & Problem-Solving using C++	✕	Fundamental of Computers	Introduction to OpenOffice Writer
II	Data Structures	✕	Cyber Security Awareness	Introduction to OpenOffice Calc
III	Digital System Design	Software Engineering	Software Engineering	Introduction to OpenOffice Impress/Cyber Security Basics
IV	Computer System Architecture	Programming in JAVA **or** C# with .NET Framework	Programming in JAVA **or** C# with .NET Framework	Unix System Administration & Shell Programming/Cyber Security advance
V	Database Management System	Programming in Python	Programming in Python	Client-side Web Technology
VI	Operating System	Data Analysis & Visualization using Python	Data Analysis & Visualization using Python	Server-side Web Technology
VII	Theory of Computation	Computer Networks Discrete Mathematics Research Methodology Mobile App Design and Development	Mobile App Design and Development	✕
VIII	Compiler Design	Design & Analysis of Algorithm Artificial Intelligence Computer Graphics Cloud Computing Web App Design and Development	Cloud Computing	✕
IX	Machine Learning	Data Mining & Warehousing Cryptography & Network Security Internet of Things Quantum Computing	Internet of Things	✕
X	Digital Image Processing	Natural Language Processing Advance Java Web Hacking and Security	Web Hacking and Security	✕

		System of Cyber Security		

Definition of Credit	
1 Hr. Lecture (L) per week	**1 Credit**
1 Hr. Tutorial (T) per week	**1 Credit**
1 Hr. Practical (P) per week	**0.5 Credit**
2 Hours Practical (P) per week	**1 Credit**

LIST OF ALL PAPERS WITH SEMESTER WISE TITLES IN "COMPUTER SCIENCE"

CERTIFICATE COURSE

Year	Semester	Course Type	Course Code	Paper Title	Credits			
					L	T	P	C
I	I	DSC 1		Computer Fundamentals & Problem-Solving using C++	3	0	1	4
		GE 1		Fundamental of Computers	3	1	0	4
	II	DSC 2		Data Structures	3	0	1	4
		GE 2		Cyber Security Awareness	3	1	0	4

DIPLOMA COURSE

Year	Semester	Course Type	Course Code	Paper Title	Credits			
					L	T	P	C
II	III	DSC 3		Digital System Design	3	1	0	4
		GE 3/DSE 1		Software Engineering	3	1	0	4
	IV	DSC 4		Computer System Architecture	3	1	0	4
		GE 4/DSE 2		Programming in JAVA	3	0	1	4
		GE 5/DSE 3		C# with .NET Framework	3	0	1	4

BACHELOR

Year	Semester	Course Type	Course Code	Paper Title	Credits			
					L	T	P	C
III	V	DSC 5		Database Management System	3	0	1	4
		GE 6/DSE 4		Programming in Python	3	0	1	4
	VI	DSC 6		Operating System	3	1	0	4
		GE 7/DSE 5		Data Analysis & Visualization using Python	3	0	1	4
		IAPC		Internship/Apprenticeship / Project/ Community Outreach				2

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Year	Semester	Course Type	Course Code	Paper Title	Credits			
					L	T	P	C
IV	VII	DSC 7		Theory of Computation	3	1	0	4
		DSE 6		Computer Networks	3	1	0	4
		DSE 7		Discrete Mathematics	3	1	0	4
		DSE 8		Research Methodology	3	1	0	4
		DSE 9/GE 8		Mobile App Design and Development	3	0	1	4
		DISSERTATION 1		DISSERTATION on MAJOR				6
	VIII	DSC 8		Compiler Design	3	1	0	4
		DSE 10		Design & Analysis of Algorithm	3	0	1	4
		DSE 11		Artificial Intelligence	3	1	0	4
		DSE 12		Computer Graphics	3	1	0	4
		DSE 13		Web App Design and Development	3	0	1	4
		DSE 14/GE 9		Cloud Computing	3	1	0	4
		DISSERTATION 2		DISSERTATION on MAJOR				6

MASTER'S IN COMPUTER SCIENCE

Year	Semester	Course Type	Course Code	Paper Title	Credits			
					L	T	P	C
V	IX	DSC 9		Machine Learning	3	0	1	4
		DSE 15		Data Mining & Warehousing	3	1	0	4
		DSE 16		Cryptography & Network Security	3	1	0	4
		DSE 17		Quantum Computing	3	0	1	4
		DSE 18/GE 10		Internet of Things	2	0	2	4
		DISSERTATION 3		DISSERTATION on MAJOR				6
	X	DSC 10		Digital Image Processing	3	0	1	4
		DSE 19		Natural Language Processing	3	0	1	4
		DSE 20		Advance Java	3	0	1	4
		DSE 21		System of Cyber Security	3	0	1	4
		DSE 22/GE 11		Web Hacking and Security	3	0	1	4
		DISSERTATION 4		DISSERTATION on MAJOR				6

Programme outcomes (POs):	
PO 1	Gain a complete exposure to the theories and practices of Computer science.
PO 2	Get transformed into a skilled learner and active programmer, enabling the students to focus on their higher studies.
PO 3	Value computer professionals and programmers.
PO 4	Explore how the concepts and applications of Computer science led to innovative thinking with a problem-solving attitude.
Programme specific outcomes (PSOs): Certificate Course	
PSO 1	Develop a strong foundation in computer fundamentals, including hardware, software, and operating systems, enabling students to understand and operate various computing environments effectively.
PSO 2	Cultivate proficiency in C++ programming and problem-solving techniques, allowing students to design, implement, and test algorithms to solve real-world problems.
PSO 3	Gain expertise in data structures, enhancing the ability to organize, manage, and analyze data efficiently for optimized performance in computational tasks.
PSO 4	Acquire practical skills in using OpenOffice Writer and Calc for creating, editing, and managing documents and spreadsheets, equipping students with essential tools for academic and professional productivity.
PSO 5	Develop a comprehensive understanding of cyber security principles and practices, enabling students to recognize, prevent, and respond to various cyber threats, ensuring safe and secure digital interactions.
Programme specific outcomes (PSOs): Diploma Course	
PSO 1	Understand and design digital systems and computer architectures, providing a solid foundation in the principles and practices of hardware and system-level design.
PSO 2	Develop comprehensive skills in software engineering methodologies and practices, including software design, development, testing, and maintenance, ensuring the creation of reliable and efficient software solutions.
PSO 3	Gain proficiency in modern programming languages, such as Java and C# with .NET Framework, enabling students to develop robust and scalable applications across various platforms.
PSO 4	Master the use of OpenOffice Impress for creating and delivering effective presentations, enhancing communication and presentation skills for academic and professional settings.
PSO 5	Acquire advanced skills in Unix system administration and shell programming, empowering students to manage Unix-based systems and automate tasks through scripting for improved productivity and system efficiency.
Programme specific outcomes (PSOs): Bachelor in Science (with specialization in Computer Science)	
PSO 1	Develop a comprehensive understanding of database management systems, including database design, implementation, and administration, enabling students to efficiently manage and manipulate data.
PSO 2	Gain proficiency in Python programming and its applications in data analysis and visualization, equipping students with the skills to analyze and interpret data for informed decision-making.
PSO 3	Acquire expertise in client-side and server-side web technologies, allowing students to design, develop, and deploy dynamic and responsive web applications.
PSO 4	Understand the fundamental concepts and functionalities of operating systems, including process management, memory management, and file systems, ensuring effective system-level programming and administration.
PSO 5	Master data visualization techniques using Python, empowering students to create insightful and interactive visual representations of data to communicate complex information effectively.

Programme specific outcomes (PSOs):Bachelor of Computer Science (HONOURS)	
PSO 1	Develop a deep understanding of the theoretical foundations of computer science, including theory of computation and discrete mathematics, enabling students to solve complex computational problems and understand the limits of computation.
PSO 2	Gain comprehensive knowledge of computer networks and cloud computing, equipping students with the skills to design, implement, and manage networked and cloud-based systems for enhanced connectivity and resource management.
PSO 3	Acquire expertise in mobile and web app design and development, as well as compiler design, preparing students to create efficient, user-friendly, and robust software applications across various platforms.
PSO 4	Master the design and analysis of algorithms and artificial intelligence techniques, allowing students to develop optimized solutions and intelligent systems for real-world applications.
PSO 5	Enhance research methodology skills, enabling students to conduct rigorous research and contribute to academic and industrial advancements, while completing a major dissertation that demonstrates their ability to undertake and present significant research projects.
PSO 6	Gain proficiency in computer graphics and data visualization, empowering students to create visually compelling graphics and effectively communicate complex data through interactive visual representations.
Programme specific outcomes (PSOs): Master of Science (Computer Science)	
PSO 1	Develop expertise in machine learning, data mining, and data warehousing, enabling students to analyze large datasets, extract meaningful insights, and build predictive models for various applications.
PSO 2	Gain comprehensive knowledge of cryptography, network security, and Quantum computing, equipping students with the skills to protect information systems, ensure data privacy, and maintain security in digital communications.
PSO 3	Acquire proficiency in emerging technologies such as the Internet of Things (IoT) and digital image processing, preparing students to design and implement innovative solutions in connected and visual computing environments.
PSO 4	Master natural language processing (NLP) and advanced Java programming, enabling students to develop sophisticated applications that understand and generate human language, and create robust software solutions.
PSO 5	Understand the principles of system cybersecurity and web hacking and security, empowering students to identify vulnerabilities, implement security measures, and ensure the integrity and resilience of information systems.
PSO 6	Enhance research methodology skills, enabling students to conduct rigorous research and contribute to academic and industrial advancements, while completing a major dissertation that demonstrates their ability to undertake and present significant research projects.

Subject: Computer Science			
Programme/Class: Certificate	**Year:** 1st	**Semester:** I	**Course Type: DSC 1**
Course Code:	**Course Title:** Computer Fundamentals & Problem-Solving using C++		**Credits: L | T | P (3 | 0 | 1)**

Course outcomes:

- Bridge the fundamental concepts of computers with the present level of knowledge of the students.
- Familiarize operating systems, programming languages, peripheral devices, networking, multimedia and internet
- Understand binary, hexadecimal and octal number systems and their arithmetic.
- Understand the difference between the top-down and bottom-up approach and concepts of object-oriented programming in connection with C++.
- Illustrate the process of data file manipulations using C++ and apply virtual and pure virtual function & complex programming situations.

Course Prerequisites: Basic understanding of computer fundamentals and logic.

Course Outline:

Unit	Topics	No. of Lab/Lectures
I.	**Introduction to Computer:** Computer System, Advantages and Disadvantages of Computer System, Evolution of computers, Generation of Computers, Classification of Computers, Block Diagram of a Digital Computer, introduction to Input/ Output Devices.	6
II.	**Memory:** Memory hierarchy, Registers (Types of Registers), Cache Memory. **Primary Memory** (RAM, how data is stored in a RAM, DRAM and SRAM. ROM (BIOS/Firmware & Types of ROM). **Secondary Memory** (Hard disk: Structure of a hard disk, how data is stored in a hard disk, concept of tracks, sectors, clusters, cylinders, Various Storage Devices (Magnetic Tape, Floppy Disks, Optical Disks, SD/MMC Memory cards, USB Pen drive).	6
III.	**Software:** Software and its Need, Types of Software: - System software, Application software. **Operating System:** History of Operating System, Function of Operating System, OS classification (Batch, Multiprogramming, Multitasking, Multithreading, Multiprocessing, Multiuser, Time sharing, real time). Programming languages, **Translators:** Compiler, Interpreter and Assembler. **Network Fundamental**: Categories, Data flow, Topology.	6
IV.	**Fundamentals of C++:** Data Types and Sizes, Declaration of variables, Modifiers, Identifiers and keywords, Symbolic constants. Operators, Precedence and order of evaluation. **Control statements**: if-else, else-if clause, switch. Loops: for, while, do-while, break, continue. **Functions**: Defining a function, function prototyping and function calls, function arguments, passing by reference, inline functions, and default arguments. **Arrays**: linear arrays, multidimensional arrays, passing arrays to functions.	6
V.	**Object Oriented Concepts**: Elements of Object-Oriented programming, Objects, Classes, and OOPs features. **Classes & Objects**: Specifying a Class, Creating Objects, Accessing Class members, defining member function, Outside Member Functions as inline, Accessing Member Functions within the class, Static data member, Access Specifiers, Constructors and Destructors, Exception Handling basics	6
VI.	**Operator Overloading**: Definition, Overloadable Operators, Unary and Binary Operators overloading through Member Functions and Friend Functions, Function Overloading, Constructor Overloading. **Dynamic Memory Allocation**: Pointers to Objects, Creating and Deleting Dynamic Objects:	7

	New and Delete operators, Array of Objects, Array of Pointers to Objects, Pointers to Object Members, ‘this’ Pointer.	
VII.	Inheritance, Types of Inheritance, Virtual Functions, Pure Virtual Function, Templates, Standard Template Library, Containers: Vectors, Lists, Iterators. File Handling. Standard Template Library: STL containers containing vectors, list, queue, map, set, hash_map, hash_set. STL algorithms functions: Sorting Algorithms functions: sort, partial_sort.	8
	Lab: Computer Fundamentals & Problem-Solving using C++	15

Textbooks:

1. Rajaraman V., “Fundamentals of Computers”, Prentice-Hall of India.
2. Norton P., “Introduction to Computers”, McGraw Hill Education.
3. Goel A., “Computer Fundamentals”, Pearson.
4. James R. Rambaugh, “Object Oriented Design and Modeling”, PHI.
5. Booch Grady, “Object Oriented Analysis and Design with Application”,Pearson, 3rded.
6. Dillon and Lee, “Object Oriented Conceptual Modeling”, New Delhi PHI-1993.
7. Stephen R.Shah, “Introduction to Object Oriented Analysis and Design”, TMH.
8. Berzin Joseph, “Data Abstraction; The Object-Oriented Approach Using C++”, McGraw Hill.
9. Herbert Schildt, “C++: The Complete Reference”, McGraw Hill, 4thed., 2003.
10. Walter Savitch, “Absolute C++”, Pearson, 5thed., 2012.
11. Lipman, Stanley B, Jonsce Lajole, “C++ Primer Reading”, AWL-1999
12. Bjarne Stroustrup, “The C++ Programming Language”, Pearson, 3rded., 2002.
13. E. Balagurusamy, “Object Oriented Programming with C++”, TMH, 6thed., 2013.

Subject: Computer Science

Programme/Class: Certificate **Year:** 1st **Semester:** I **Course Type:** GE 1

Course Code: **Course Title:** Fundamental of Computers **Credits: L | T | P (3 | 1 | 0)**

Course outcomes:

- Understand different components of a computer.
- Differentiate between hardware and software.
- Learn the basic concepts of operating systems, networking, internet.
- Understand various advanced and emerging technologies.

Course Prerequisites: Basic Knowledge of Computers

Course Outline:

Unit	**Topics**	**No. of Lab/Lectures**
I.	Introduction to Computer: Definition, Computer Hardware & Computer Software Components: Hardware – Introduction, Input devices, Output devices, Central Processing Unit, Memory- Primary and Secondary. Software Introduction, Types – System and Application. Computer Languages: Introduction, Concept of Compiler, Interpreter & Assembler	12
II.	Problem solving concept: Algorithms – Introduction, Definition, Characteristics, Limitations, Conditions in pseudo-code, Loops in pseudocode. Operating system: Definition, Functions, Types, Classification, Elements of command based and GUI based operating system.	12
III.	Computer Network: Overview, Types (LAN, WAN and MAN), Data communication, topologies. Internet: Overview, Architecture, Functioning, Basic services like WWW, FTP, Telnet, Gopher etc., Search engines, E-mail, Web Browsers.	12
IV.	Internet of Things (IoT): Definition, Sensors, their types and features, Smart Cities, Industrial Internet of Things. Block chain: Introduction, overview, features, limitations and application areas fundamentals of Block Chain. Crypto currencies: Introduction, Applications and use cases Cloud Computing: Its nature and benefits, AWS, Google, Microsoft & IBM Services	12
V.	Emerging Technologies: Introduction, overview, features, limitations and application areas of Augmented Reality, Virtual Reality, Grid computing, Green computing, Big data analytics, Quantum Computing and Brain Computer Interface	12

Textbooks:

1. Rajaraman V., “Fundamentals of Computers”, Prentice-Hall of India.
2. Norton P., “Introduction to Computers”, McGraw Hill Education.
3. Goel A., “Computer Fundamentals”, Pearson.
4. Balagurusamy E., “Fundamentals of Computers”, McGraw Hill
5. Thareja R., “Fundamentals of Computers”, Oxford University Press.
6. Bindra J., “The Tech Whisperer- on Digital Transformation and the Technologies that Enable it”, Penguin

Subject: Computer Science

Programme/Class: Certificate	**Year:** 1st	**Semester:** II	**Course Type:** DSC 2
Course Code:	**Course Title:** Data Structures		**Credits: L | T | P (3 | 0 | 1)**

Course outcomes:

- Understand different problem-solving techniques
- Differentiate between sequential lists and linked lists.
- Learn and implement searching and sorting techniques.
- Understand the fundamentals and application of trees.
- Describe the graph terminologies and applications.

Course Prerequisites: Proficiency in a programming language like C++ or Java.

Course Outline:

Unit	Topics	No. of Lab/Lectures
I.	Introduction to problem solving approach: Algorithmic solution, analysis of algorithms– space and time complexity, asymptotic analysis, step counting and time complexity analysis.	9
II.	Sequential Lists and Linked Lists: Sequential lists, arrays– single and multi-dimensional arrays, sparse matrix, algorithm to store sparse matrices, singly, doubly and circular linked lists, list traversal algorithms, stacks– array implementation and linked list implementation, applications of stack, queues– array implementation and linked list implementation, circular queue and dequeue.	9
III.	Searching and Sorting: Searching algorithms– linear search, binary search, comparison of linear and binary search, constant time search using hashing, hash functions, collision resolution techniques– linear probing and chaining, Sorting algorithms– bubble sort, selection sort, insertion sort, merge sort, quick sort, radix sort, shell sort, bucket sort, comparison of sorting techniques, priority queues, binary heap, heapsort.	9
IV.	Trees: Basic terminology, tree traversals, expression trees, post/pre/infix notation, binary search tree, search, insertion and deletion operations in BST, balanced BST, AVL tree, insertion and deletion in AVL tree.	9
V.	Graphs: Graph theory terminology, graph representation, graph traversal algorithms, Breadth First Search and connected components, Depth First Search and strongly connected components, applications of BFS and DFS.	9
	Lab: Data Structures	**15**

Textbooks:

1. Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum, "Data Structures using C and C++", 2nded., Pearson Education, 2006.
2. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, "Fundamentals of Data Structures in C", Universities press, 2nded., 2008.
3. Robert Sedgewick, Kevin Wayne, "Algorithms", Pearson Education, 4thed., 2011.
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", PHI, 3rded., 2010.
5. Seymour Lipschutz, "Data Structures", Schaum's outlines, McGraw Hill Education, 1sted., 2014.
6. Donald E. Knuth, "The Art of Computer Programming", Vol. 1 and Vol. 3.

Subject: Computer Science

Programme/Class: Certificate	**Year:** 1st	**Semester:** II	**Course Type: GE 2**
Course Code:	**Course Title:** Cyber Security Awareness	**Credits: L | T | P (3 | 1 | 0)**	

Course outcomes:

- Understand the fundamentals of cyber security and its importance.
- Identify common cyber threats and vulnerabilities.
- Learn best practices for cyber security and risk management.
- Understand relevant cyber laws and ethical considerations.
- Develop practical skills to protect against cyber-attacks.

Course Prerequisites: Basic Knowledge of Computers

Course Outline:

Unit	Topics	No. of Lab/Lectures
I.	**Introduction to Cyber Security:** Definition and scope of cyber security: Understanding the need for cyber security in today's digital world and its role in protecting individuals, organizations, and governments from cyber threats. Cyber security principles and goals: Exploring the key principles of cyber security, such as confidentiality, integrity, availability, and non-repudiation.	**7**
II.	**Cyber Threats and Attacks** Common cyber threats: Studying various types of cyber threats, including malware (viruses, worms, ransomware), phishing attacks, and DDoS (Distributed Denial of Service) attacks. Understanding attack vectors and methods: Exploring how cyber-attacks are launched, including social engineering techniques and exploitation of software vulnerabilities.	**7**
III.	**Cyber Security Best Practices** Password security and management: Learning how to create strong passwords, securely store them, and use password managers. Secure browsing habits: Understanding safe browsing practices, including recognizing and avoiding suspicious websites and links. Data protection and encryption: Exploring the importance of data protection, including encryption techniques to secure data in transit and at rest.	**7**
IV.	**Cyber Security Tools and Technologies** Introduction to firewalls, antivirus software, and intrusion detection systems: Understanding the role of these tools in protecting against cyber threats and how to configure and use them effectively. Overview of encryption technologies: Exploring encryption algorithms and protocols used to secure data communication and storage.	**8**
V.	**Cyber Security Policies and Compliance** Overview of relevant laws and regulations: Studying key cyber security laws and regulations, such as GDPR, HIPAA, and the Cybersecurity Information Sharing Act (CISA). Importance of compliance and best practices: Understanding the importance of complying with cyber security regulations and implementing best practices to protect against legal and financial consequences.	**8**
VI.	**Cyber Laws and Ethics** Overview of cyber laws and regulations: Studying cyber laws related to data protection, privacy, intellectual property, and cybercrime. Understanding legal aspects of cyber security: Exploring legal implications of cyber-attacks, data breaches, and non-compliance with cyber security regulations. Ethical considerations in cyber security: Discussing ethical issues related to cyber security, such as responsible disclosure of vulnerabilities and ethical hacking.	**8**
	Tutorial:	**15**

	• Hands-on Cyber Security Practices: Installing and configuring antivirus software: Hands-on experience with installing, configuring, and updating antivirus software to protect against malware. • Implementing basic firewall rules: Configuring firewall rules to filter network traffic and protect against unauthorized access. • Conducting phishing simulations and awareness training: Practical exercises to recognize phishing emails and educate users about phishing threats.	

Textbooks:

1. "Cyber Security Essentials" by James Graham
2. "Cybersecurity for Beginners" by Raef Meeuwisse
3. Relevant legal texts and case studies

Software Requirement:

- Virtualization software (e.g., VirtualBox, VMware) for tutorial activities

Subject: Computer Science			
Programme/Class: Diploma	**Year:** 2nd	**Semester:** III	**Course Type: DSC 3**
Course Code:	**Course Title:** Digital System Design	**Credits: L | T | P (3 | 1 | 0)**	

Course outcomes:

- Understand the Boolean expressions and their realizations.
- Design combinational and sequential building blocks.
- Use these building blocks to design digital circuits.
- Learn Verilog to design/model digital system.

Course Prerequisites: Basic Knowledge of Computers

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Number Systems: Digital Computer, Number Systems– Number Representation, Binary, Octal, Hexadecimal, Unsigned and Signed Numbers, Arithmetic Operations, Fixed point and Floating-Point representations, Use of different number systems in digital design, Binary Codes– BCD, EBCDIC, ASCII, Unicode, Gray codes, Excess-3, Error Detection and Correction codes.	10
II.	Boolean Algebra and Digital Logic: Boolean Algebra, Truth Tables, Logic Gates– AND, OR, NOT, NAND, NOR, XOR, Digital Circuit Characterization– Fan-in/Fan-out, Switching Functions, Boolean Functions– Sum of Product and Product of Sum, Karnaugh Maps, Minimization of Boolean Functions, K-Maps with Don't Care, Multiple Output Functions.	10
III.	Combinational Logic & Circuit Design: Combinational Circuits– Analysis and Design Procedures, Circuits for Arithmetic Operations– Code Conversion, Binary Adder, Binary Subtractor, Decimal Adder, Magnitude Comparator, Decoders and Encoders, Multiplexers and Demultiplexers, Introduction to HDL– HDL Models of Combinational circuits, Introduction to VHDL and Basic VHDL Modelling.	10
IV.	Sequential Logic & Circuit Design: Sequential Elements– Latches and Flip Flops– Analysis and Design Procedures, Application of Flip Flops– Clock Generation, Counters, Registers, Shift Registers, State Machine Concepts– State Diagram, State Table, State Assignment and State Reduction/Minimization, HDL for Sequential Logic Circuits. Asynchronous Sequential Logic– Analysis and Design of Asynchronous Sequential Circuits, Reduction of State and Flow Tables, Race-free State Assignment, Hazards.	10
V.	Memory & Programmable Logic Devices: Memory hierarchy, Memory technologies– Cache memory, Virtual memory, TLBs, Design of memory– ROM and RAM, Programmable Logic Array (PLA), Programmable Array Logic (PAL). Different Logic families– TTL, ECL, MOS, CMOS– operation, design and specification.	10
VI.	Input-Output Organization: Peripheral Devices, I/O Modules, Isolated vs. Memory-Mapped I/O, Asynchronous Data Transfer, Modes of Transfer– Programmed I/O, Interrupt-Driven I/O, Direct Memory Access (DMA) controller, I/O Processors (IOP).	10

Textbooks:

1. M. Morris Mano, Digital Logic and Computer Design, Pearson Education, 1sted., 2004.
2. M. Morris Mano, Michael D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, Pearson Education, 5thed., 2014.

References:

1. David A. Patterson, John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann, 5thed., 2016.
2. M. Morris Mano, Computer System Architecture, Pearson Education, 3rded., 2008.
3. John F. Wakerly, Digital Design Principles and Practices, Pearson Education, 4thed., 2007.
4. Charles H. Roth Jr, Fundamentals of Logic Design, Jaico Publishing House, 5thed., 2003.
5. 5. Donald D. Givone, Digital Principles and Design, Tata McGraw Hill, 2003.

Subject: Computer Science			
Programme/Class: Diploma	**Year:** 2nd	**Semester:** III	**Course Type: GE 3/DSE 1**
Course Code:	**Course Title:** Software Engineering	**Credits: L | T | P (3 | 1 | 0)**	

Course outcomes:

- Understand the terminologies of software engineering paradigms
- Describe the software engineering requirements and metrics
- Analyze the software development life cycle.
- Explain the software maintenance and current trends in software engineering
- Demonstrate the Computer Aided Software Engineering (CASE) tools

Course Prerequisites: Basic programming skills and understanding of software development concepts.

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Software Engineering Paradigms: Software Characteristics, Software myths, Software Applications, Software Engineering Definitions, Software Process Models, Waterfall, Prototyping, Spiral (including WIN-WIN Spiral), RAD, Process iteration, Process activities, Software Project Management: Management activities, Project planning, Project scheduling, Risk management and activities.	10
II.	Software Requirements Engineering: Requirements Engineering Processes, Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management. Software Requirements, Functional and non-functional requirements, User requirements, System requirements, Interface specification, Software Requirement Specification (SRS) document. Specification languages. Software Metrics and Measures—Process Metrics, Project metrics, Software Project Estimation Models- Empirical, Putnam, COCOMO models.	10
III.	Software Design Process: Principles of software design, Design Strategies, Levels of software design, Interface Design, Coding, Software Reuse. Software Testing, Software Reliability, Software Safety, Defect testing, Debugging Tools.	10
IV.	Maintenance: Types of Maintenance, Maintenance Cost, Software Configuration Management, Software Reuse, Software Evolution, Reverse Engineering, Introduction to legacy systems, Software Quality Assurance– plans & activities, Software Documentation. Role of documentation in maintenance and types of documentation.	10
V.	Current trends in Software Engineering: Software Engineering for projects & products. Introduction to Web Engineering and Agile process.	10
VI.	CASE Tools: Computer Aided Software Engineering (CASE), Introduction to CASE tools, Types of CASE tools, Project Management Tools, Analysis tools, Design tools, Programming tools, Prototyping tools, Maintenance tools, Advantages and disadvantages of CASE tools.	10

Textbooks:

1. K.K. Aggarwal, Yogesh Singh, "Software Engineering", New Age International, 2nded., 2005.
2. R.S. Pressman, "Software Engineering – A practitioner's approach", McGraw Hill, 5thed., 2001.
3. Stephen R. Schach, "Classical & Object Oriented Software Engineering", IRWIN, 1996
4. James Peter, W. Pedrycz, "Software Engineering: An Engineering Approach", John Wiley & Sons.
5. I. Sommerville, "Software Engineering", Addison Wesley, 2002.

Subject: Computer Science			
Programme/Class: Diploma	**Year:** 2nd	**Semester:** IV	**Course Type: DSC 4**
Course Code:	**Course Title:** Computer System Architecture	**Credits: L | T | P (3 | 1 | 0)**	

Course outcomes:

- To develop understanding of Computer Models and its usage.
- To develop understanding of ALU Design.
- To conceptualize the understanding of Control Unit design, Memory, IPC, Control Design.
- To develop understanding of Memory & Input/output organization Overview.

Course Prerequisites: Basic understanding of computer organization.

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Register Transfer and Microoperations: Components of a computer system, Von Neumann architecture, Computer System Interconnection, Register Transfer Language, Register Transfer, Bus and Memory Transfers, Microoperations– Arithmetic, Logic and Shift.	15
II.	Central Processing Unit: Computer Arithmetic– ALU, Integer Representation and Arithmetic, Floating-Point Representation and Arithmetic, Decimal Arithmetic, CU Implementation– Hardwired and Multi Programmed, Multiplier Control Unit, CPU Control Unit, Instruction Set Architecture– Addressing Modes and Design, CISC and RISC paradigm, Basic MIPS implementation– Building data path– Control Implementation scheme.	15
III.	The 8086 microprocessor: Introduction to 8086 – Microprocessor architecture – Addressing modes, Instruction set and assembler directives- 8086 signals – Basic configurations – System bus timing –System design using 8086- System Bus Structure-Memory Interfacing, I/O interfacing, Parallel communication interface – Serial communication interface – D/A and A/D Interface.	15
IV.	Parallel Processing concepts: Instruction level parallelism, Parallel processing challenges, Flynn's classification, Pipelining, Vector Processing, Superscalar processors, Multi-core Processors– Multithreading, Multicore processor Architecture, Multiprocessor configurations – Coprocessor, closely coupled and loosely coupled configurations, Cache Coherence Protocols, Synchronization, Memory Consistency.	15

Textbooks:

1. M. Morris Mano, "Computer System Architecture", Pearson, 3rded., 2007.
2. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", McGrawHill, 5thed., 2002.
3. Yu-Cheng Liu, Glenn A. Gibson, "Microcomputer Systems: The 8086 / 8088 Family: Architecture, Programming and Design", Prentice Hall of India, 2nded., 2007.

References:

1. David A. Patterson, John L. Hennessy, "Computer Organization and Design: The Hardware/Software Interface", Morgan Kaufmann, 3rded., 2007.
2. John L. Hennessy, David A. Patterson, "Computer Architecture: A Quantitative Approach", Morgan Kaufmann, 5thed., 2012.
3. John P. Hayes, "Computer Architecture and Organization", McGraw-Hill Education, 2nded., 1998.
4. William Stallings, "Computer Organization and Architecture – Designing for Performance", Pearson Education, 6thed., 2003.
5. Doughlas V. Hall, "Microprocessors and Interfacing, Programming and Hardware", TMH, 2012.

Subject: Computer Science			
Programme/Class: Diploma	**Year:** 2^{nd}	**Semester:** IV	**Course Type: GE 4/DSE 2**
Course Code:	**Course Title:** Programming in JAVA	**Credits: L | T | P (3 | 0 | 1)**	

Course outcomes:

- Use the syntax and semantics of java programming language and basic concepts of OOP.
- Develop reusable programs using the concepts of inheritance, polymorphism, interfaces and packages.
- Apply the concepts of Multi-threading and Exception handling to develop efficient and error free codes.
- Design event driven GUI Applications.

Course Prerequisites: Proficiency in basic programming concepts.

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	Features of java, JDK Environment & tools like (java, javac, applet viewer, javadoc, jdb), OOPs Concepts Class, Abstraction , Encapsulation, Inheritance, Polymorphism, Difference between C++ and JAVA, Structure of java program, Data types ,Variables ,Operators , Keywords ,Naming Convention, Decision Making (if, switch), Looping(for, while), Type Casting, Array Creating an array Types of Array - One Dimensional arrays - Two Dimensional array, String - Arrays , Methods. – String Buffer class	9
II.	Creating Classes and objects, Memory allocation for objects, Constructor, Implementation of Inheritance Simple, Multilevel, Interfaces, Abstract classes and methods, Implementation of Polymorphism, Method Overloading, Method Overriding, Nested and Inner classes, Modifiers and Access Control, Packages Packages Concept Creating user defined packages, Java Built in packages: java.lang->math, java.util->Random, Date, Hashtable, Wrapper classes	9
III.	Collection Framework, Interfaces - Collection - List - Set - SortedSet - Enumeration - Iterator – ListIterator, Classes - LinkedList - ArrayList - Vector - HashSet	9
IV.	Exception: Exception types, Using try catch and multiple catch Nested try, throw, throws and finally, Creating user defined Exceptions File Handling: Stream ByteStream Classes CharacterStream Classes, File IO basics, File operations Creating file Reading file (character, byte) Writing file (character, byte), MultiThreading	9
V.	AWT: Components and container used in AWT, Layout managers, Listeners and Adapter classes,Event Delegation model, Swing: Introduction to Swing Component and Container Classes	9
	Lab: Programming in JAVA	15

Textbooks:

1. Herbert Schildt, “Java, The Complete Reference”, TMH, 7thed., 2007.
2. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4thed., 2005.
3. Cay S. Horstmann, GaryCornell, "Core Java 2 Volume 1 and 2”, Prentice Hall, 9thed., 2012.
4. Bruce Eckel, "Thinking in Java", PHI, 3rded., 2002.
5. Paul Deitel, Harvey Deitel, "Java: How to Program", Prentice Hall, 10thed., 2011.
6. Bert Bates and Kathy Sierra "Head First Java", Orielly Media Inc., 2nded., 2005.
7. Elliotte Rusty Harold, “Java Network Programming”, O’Reilly publishers, 2000
8. 8. Ed Roman, “Mastering Enterprise Java Beans”, John Wiley & Sons Inc., 1999.

Subject: Computer Science			
Programme/Class: Diploma	**Year:** 2nd	**Semester:** IV	**Course Type: GE 5/DSE 3**
Course Code:	**Course Title:** C# with .NET Framework	**Credits: L | T | P (3 | 0 | 1)**	

Course outcomes:

- Acquire the knowledge of the structure and model of the programming language C #
- Understand the use of programming language C # for various programming technologies
- Evaluate user requirements for software functionality required to decide whether the programming language C # can meet user requirements
- Develop variety of software in C #

Course Prerequisites: Proficiency in basic programming concepts.

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	The .NET Framework: Introduction, Common Language Runtime, Common Type System, Common Language specification, The Base Class Library, The .Net class library Intermediate language, Just-in time Compilation, Garbage Collection, Application Installation and Assemblies, Web services, Unified classes.	9
II.	C# Basics: Introduction, Data Types, Identifiers, Variables and constants, C# statements, Object Oriented Concept, Object and Classes, Arrays and Strings, System collections, Delegates and Events, Indexes, Attributes, versioning.	9
III.	C# Using Libraries: Namespace- System, Input Output, Multi-Threading, Networking and Sockets, Data Handling, Windows Forms, C# in web application, Error Handling	9
IV.	Advanced Features Using C#: Web services, Windows services, messaging, Reflection, COM and C#, Localization.	9
V.	Advanced Features Using C#: Distributed Application in C#, XML and C#, Unsafe Mode, Graphical Device Interface with C#, CASE Study (Messenger Application)	**9**
	Lab: C# with .NET Framework	**15**

Textbooks:

1. Jeffrey Richter, “Applied Microsoft .NET Framework Programming”, (Microsoft)
2. Fergal Grimes, “Microsoft .Net for Programmers”, (SPD)
3. Balagurusamy, “Programming with C# “, TMH
4. Wiley,” Beginning Visual C# 2008”,Wrox

<table>
<tr><td colspan="4"><b>Subject:</b> Computer Science</td></tr>
<tr><td><b>Programme/Class:</b> B.Sc.</td><td><b>Year:</b> 3rd</td><td><b>Semester:</b> V</td><td><b>Course Type: DSC 5</b></td></tr>
<tr><td><b>Course Code:</b></td><td><b>Course Title:</b> Database Management System</td><td colspan="2"><b>Credits: L | T | P (3 | 0| 1)</b></td></tr>
</table>

Course outcomes:

- Demonstrate a clear understanding of the basics of Database and its use.
- Implement Relational Model for Industry as well for all organizations
- Understanding Normalization for fast access of records as well transactions

Course Prerequisites: Basic understanding of databases and SQL.

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	Introduction: Overview of databases, Data models, DBMS architecture and data independence, History of Database Systems. Entity-Relationship Modeling: Basic concepts, constraints, keys, Design issues, weak entities, enhanced E-R, Sub Classes, Super classes, inheritance, specialization and generalization.	**9**
II.	Relational Data Model and Normalization: Relational model concepts, relational constraints, relational algebra, relational calculus. SQL: basic queries, nested subqueries, aggregate functions, null values, complex queries, database modification commands, programming using SQL, embedded SQL, dynamic SQL. Database Design– Functional dependencies, Normalization, Normal form– 1NF, 2NF, 3NF, BCNF.	**9**
III.	Integrity and Security: Domain Constraints, Referential Integrity Constraints, Assertions, Triggers, Security and Authorization– Authorization in SQL, Encryption and Authentication.	**9**
IV.	File Organization: Indexed sequential access files, implementation using B+ trees, hashing, hashing functions, collision resolution, extendible hashing, dynamic hashing approach implementation and performance.	**9**
V.	Transaction and Concurrency Control: Transaction concept, Transaction state, ACID properties and their implementation. Concurrency Control– Lock Based Protocols, Timestamp Based Protocols, Validation Based Protocols, Multiple Granularity. Recovery System– Failure Classification, Storage Structure, Recovery and Atomicity, Log based recovery.	**9**
	Lab: Database Management System	**15**

Textbooks:

1. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Pearson Education, 5thed., 2008.
2. 2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database Concepts”, McGraw-Hill, 6thed., 2013.
3. R. Ramakrishanan, J. Gehrke, “Database Management Systems” McGraw-Hill, 3rded., 2002.
4. Peter Rob, Carlos Coronel, “Database Systems: Design, Implementation and Management”, 7th ed., 2006.

Subject: Computer Science			
Programme/Class: B.Sc.	**Year:** 3rd	**Semester:** V	**Course Type: GE 6/DSE 4**
Course Code:	**Course Title:** Programming in Python	**Credits: L | T | P (3 | 0| 1)**	

Course outcomes:

- Understand the fundamentals of Python programming language.
- Learn to connect Python with databases.
- Develop skills to interact with databases using Python.
- Gain practical experience in developing database-driven applications.

Course Prerequisites: Proficiency in basic programming concepts.

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	**Introduction to Python:** History and features of Python, Setting up Python environment (interpreter, IDEs) Basic Python Programming: Variables, data types, and operators, Control flow (if-else, loops), Functions and modules	**9**
II.	**Data Structures in Python:** Lists, tuples, dictionaries, Sets and their operations, Working with files	**9**
III.	**Object-Oriented Programming in Python:** Classes and objects, Inheritance and polymorphism, Encapsulation and abstraction	**9**
IV.	**Advanced Python Programming:** Exception handling, Decorators and generators, Regular expressions	**9**
V.	**Database Connectivity in Python:** Overview of database concepts, Connecting Python with databases (SQLite, MySQL, PostgreSQL), Performing CRUD operations (Create, Read, Update, Delete) using Python	**9**
	Lab: Programming in Python	**15**

Textbooks:

1. "Python Crash Course" by Eric Matthes
2. "Fluent Python" by Luciano Rama

Subject: Computer Science

Programme/Class: B.Sc. **Year:** 3rd **Semester:** VI **Course Type: DSC 6**

Course Code: **Course Title:** Operating System **Credits: L | T | P (3 | 1| 0)**

Course outcomes:

- Understand concept of different type of Operating Systems
- Understand the Program, Processes difference and used corresponding to devices.
- Understand the efficient memory utilization with file management

Course Prerequisites: DSC 1/DSC 3/DSE 4

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	Operating System Overview: Operating Systems– objectives and functions, evolution– early Operating Systems, Parallel systems, Distributed Systems, Process Control & Real-time Systems. Modern Operating Systems, Virtual Machines, OS Design considerations for Multiprocessor and Multicore architectures. OS Organization– Processor and user modes, Kernel, System Calls, System Programs, System Boot, Overview and Booting process of various OS– Microsoft Windows, Modern UNIX, Linux, Android.	12
II.	Process Management: System view of the process and resources, process abstraction, process hierarchy, process control, execution of the OS. Threads– concept, issues, libraries, thread programming using pthread, multicore processors and threads, multithreading models, process and thread management in Linux, Android and Windows. Process scheduling– Uniprocessor, Multiprocessor and Real-time scheduling algorithms, Traditional UNIX scheduling, Linux scheduling. Concurrency– Process/Thread synchronization, Mutual Exclusion Principles of Concurrency, Critical Section Problem, Hardware support, OS support (semaphores, mutex), Programming Language support (monitors), Classical synchronization problems– Readers/Writers problem, Producer/Consumer problem. Deadlocks– Deadlock characterization, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Dining Philosophers Problem. Linux and Android interprocess communication (IPC) and concurrency mechanisms.	18
III.	Memory Management: Logical vs. Physical Address space, Memory Partitioning– Fixed and Dynamic Partitioning, Buddy System, Relocation, Paging, Segmentation. Virtual Memory– Demand Paging, Page Replacement, Frames allocation, Thrashing, Allocating Kernel Memory. Memory Management in Linux.	15
IV.	I/O and File Management: I/O Devices, Buffering, Disk Scheduling, Sector Queuing, Linux I/O. File– File Concept, File Organization, Access Methods, File Sharing and Protection, Logical and Physical File System, Directory Structure, Allocation Methods– Contiguous, Sequential and Indexed Allocation, Linux Virtual File System. Case Study: Linux and Windows Operating Systems.	15

Textbooks:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, “Operating System Concepts”, John Wiley Publications, 8thed., 2008.
2. William Stallings, “Operating Systems: Internals and Design Principles”, Pearson, 7thed., 2013.
3. Robert Love, “Linux Kernel Development”, Pearson, 1sted., 2010.

References:

1. Dhananjay M. Dhamdhere, “Systems Programming and Operating Systems”, Tata McGraw-Hill, 2nded., 1999.
2. Gary Nutt, “Operating Systems: A Modern Perspective”, Pearson, 3rded., 2009.
3. Maurice J. Bach, “The Design of the UNIX Operating System”, PHI.

Subject: Computer Science			
Programme /Class: B.Sc.	**Year:** 3rd	**Semester:** VI	**Course Type: GE 7/DSE 5**
Course Code:	**Course Title:** Data Analysis & Visualization using Python		**Credits: L | T | P (3 | 0| 1)**

Course outcomes:

- Apply descriptive statistics to obtain a deterministic view of data
- Perform data handling using Numpy arrays
- Load, clean, transform, merge and reshape data using Pandas
- Visualize data using Pandas and matplot libraries

Course Prerequisites: Proficiency in Python programming.

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Introduction to basic statistics and analysis: Fundamentals of Data Analysis, Statistical foundations for Data Analysis, Types of data, Descriptive Statistics, Correlation and covariance, Linear Regression, Statistical Hypothesis Generation and Testing, Python Libraries: NumPy, Pandas, Matplotlib	11
II.	Array manipulation using Numpy: Numpy array: Creating numpy arrays, various data types of numpy arrays, indexing and slicing, swapping axes, transposing arrays, data processing using Numpy arrays	11
III.	Data Manipulation using Pandas: Data Structures in Pandas: Series, DataFrame, Index objects, loading data into Pandas data frame, Working with DataFrames: Arithmetics, DRAFTStatistics, Binning, Indexing, Reindexing, Filtering, Handling missing data, Hierarchical indexing, Data wrangling: Data cleaning, transforming,merging and reshaping	12
IV.	Plotting and Visualization: Using matplotlib to plot data: figures, subplots, markings, color and line styles, labels and legends, Plotting functions in Pandas: Line,bar, Scatter plots, histograms, stacked bars, Heatmap	11
	Lab: Data Analysis & Visualization using Python	15

Textbooks:

1. "Python for Data Analysis" by Wes McKinney
2. "Interactive Data Visualization with Python" by Bokeh Development Tea

Software Requirement:

1. Python (latest version)
2. Jupyter Notebook (for coding and documentation)
3. Required libraries (pandas, NumPy, matplotlib, Seaborn, Plotly)

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VII	**Course Type: DSC 7**
Course Code:	**Course Title:** Theory of Computation	**Credits: L | T | P (3 | 1| 0)**	

Course outcomes:

- Understand formal languages, grammars and Chomsky hierarchy.
- Design regular grammar, DFA, NFA, Mealy and Moore machine, PDA, Turing machines.
- Understand the notion of decidability and computability.

Course Prerequisites: DSC 1

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	Finite automaton (FA): Transition system, Acceptance by a finite automaton, Deterministic and non-deterministic automaton (DFA and NFA), Equivalence of DFA and NFA, Minimization of states in a finite automaton, Mealy and Moore machines.	12
II.	Formal languages and grammar: - Chomsky's hierarchy, Regular grammar and regular expression, REs and FA, Closure Properties, Pumping lemma for regular sets.	12
III.	Push down automaton (PDA): Acceptance by PDA using final state or empty stack, Context free language, Deterministic CFL, Deterministic PDA, Context free grammar and derivation trees, Leftmost and rightmost derivation, Ambiguity in context free grammar, Pumping lemma for context free languages.	12
IV.	Turing machine (TM): TM as computable functions and accepters, Non-deterministic TM, type-0 grammar, Halting problem of a TM, Linear bounded automaton (LBA) model, Context sensitive languages and grammars.	12
V.	Unsolvable problems: Reduction techniques, Decidability– Post's correspondence problem (PCP), Rice's theorem, Decidability of membership, Emptiness and equivalence problems of languages, P, NP, NP-Completeness; Satisfiability and Cook's theorem.	12

Textbooks:

1. KLP Mishra, N. Chandrasekaran, "Theory of Computer Science (Automata, Languages and Computation)", PHI, 3rded.
2. Peter Linz, Jones, Bartlett, "An Introduction to Formal Languages and Automata", 5thed.
3. John E. Hopcroft, J.D. Ullman, Rajiv Motwani, "Introduction to Automata Theory, Languages and Computation", Pearson Education, 3rded.
4. Michael Sipser, "Introduction to the Theory of Computation", Cengage Learning, 3rded.

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VII	**Course Type: DSE 6**
Course Code:	**Course Title:** Computer Networks	**Credits: L | T | P (3 | 1| 0)**	

Course outcomes:

- Understanding Network topologies and network architecture.
- Demonstrate a clear understanding of the different layers of network architecture.

Course Prerequisites: DSC 1

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Introduction to Computer Networks and Data Communication: Network definition; network topologies; network classifications; network protocol; layered network architecture; overview of OSI reference model; overview of TCP/IP protocol suite. Analog and digital signal; data-rate limits; digital to digital line encoding schemes; pulse code modulation; parallel and serial transmission; digital to analog modulation; multiplexing techniques– FDM, TDM; transmission media.	12
II.	Networks Switching Techniques and Access mechanisms: Circuit switching; packet switching– connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.	12
III.	Data Link Layer: Error detection and error correction techniques; data-link control-framing and flow control; Error recovery protocols– stop and wait ARQ, go-back-n ARQ; CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks–repeaters, hubs, switches, bridges, router and gateways.	12
IV.	Network and Transport Layers: Routing; routing algorithms; Network layer protocol of Internet– IP protocol, Internet control protocols, Transport services, Transport protocols, Internet Transport Protocols– UDP, TCP.	12
V.	Application Layer: Client–server model, Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), Telnet, Network Virtual Terminal (NVT), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), User Agent (UA), Mail Transfer Agent (MTA), Multipurpose Internet Mail Extensions (MIME), Post Office Protocol (POP), Simple Network Management Protocol (SNMP), Hypertext Transfer Protocol (HTTP), Uniform Resource Locator (URL), World Wide Web (WWW).	12

Textbooks:

1. B.A. Forouzan, "Data Communications and Networking", THM, 4th ed., 2007.
2. Andrew S. Tanenbaum, "Computer Networks", PHI, 4th ed., 2003.

References:

1. J.F. Kurose, K.W. Ross, "Computer Networking: A Top-Down Approach", Pearson, 6th ed., 2012.
2. Leon Garcia, Widjaja, "Communication Networks: Fundamental Concepts and Key Architectures", Tata McGraw Hill, 2001.

Subject: Computer Science

Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VII	**Course Type: DSE 7**
Course Code:	**Course Title:** Discrete Mathematics	**Credits: L | T | P (3 | 1| 0)**	

Course outcomes:

- Understand and interpret the fundamental mathematical structures like Set theory, Relation and Functions
- Write recursive definitions of sequences and collections of objects
- Understand the concepts and applications of vector algebra
- Understand and interpret the basic concepts of Graph Theory
- Apply the use of graph theory concepts solving various Computer Science and Engineering problems.

Course Prerequisites: Mathematics in 10+2

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Logic and Proofs: Propositional Logic– Binary logic and propositions, Propositional Variables, Truth table, Logical connectives– Negation, Conjunction, Disjunction, Conditional, Biconditional, Universal connectives, Well-formed Formulas, Tautology, Contradiction and Contingency, Propositional Equivalences, Duality, Predicate Logic– Predicates, Quantifiers– Existential and Universal quantifier, Predicate formulas, Equivalence of formulas involving quantifiers, Normal forms– CNF/DNF, PCNF/PDNF, Normal forms for First Order Logic– Prenex Normal Form, Rules of Inference. Proof Techniques– Introduction to Proof, Definitions– Theorem, Lemma, Corollary and Conjecture, Methods of Proof– Direct Proofs, Indirect Proofs– Proof by Contraposition, Proof by Contradiction, Proof by Cases, Mathematical Induction.	12
II.	Counting and Recurrence: Set Theory, Countable and Countably Infinite Sets, Pigeonhole Principle, Permutation and Combination, Principle of Inclusion and Exclusion, Generating functions– Definition, Generating Permutations and Combinations. Recurrence– Recurrence Relations, Linear Recurrence Relations with constant coefficients and their solution, Solving Linear Recurrence Relations using Generating Functions.	12
III.	Binary and Ordered Relations: Binary Relation, Properties of Binary Relations– Reflexive, Symmetric and Transitive Relation, Equivalence Relation, Closure of Relations– Reflexive, Symmetric and Transitive Closure, Warshall's algorithm, Ordered Relation– Partial Order and Posets, Hasse diagram of Poset, Maximal, Minimal, Maximum and Minimum of poset, glb and lub, Isomorphic ordered set, Well ordered set, Lattice, Properties of lattice, Distributed and Complemented lattice, Applications of Lattice, Topological Sort.	12
IV.	Graph Theory: Definition of Graph, Types– Directed and Undirected Graph, Complete Graph, Bipartite Graph, Multigraph, Weighted Graph, Graph Representation– Adjacency matrix and Adjacency list, Graph Isomorphism, Connectivity and Path, Euler and Hamiltonian Paths and Circuits, Shortest path– Dijkstra's algorithm, Planar Graph, Euler's theorem for Planar Graphs, Graph Coloring. Trees– Basic terminology and properties, Tree Traversal– Inorder, Preorder and Postorder, Expression Trees– Infix, Prefix and Postfix notations, Spanning Trees– Kruskal's and Prim's algorithms for Minimum Spanning Trees (MST).	12
V.	Algebra: Definition and elementary properties of Semigroups, Monoids, Groups, Subgroups, Generators and Cyclic group, Permutation group, Cosets, Lagrange's Theorem, Rings, Integral Domains and Fields	12

Textbooks:

1. Kenneth H. Rosen, "Discrete Mathematics and Its Applications", Tata McGraw Hill, 7th ed., 2012.

2. C. L. Liu, “Elements of Discrete Mathematics”, McGraw Hill, 2nd ed., 1986.
3. Bernard Kolman, Robert C. Busby, Sharon Cutler Ross, “Discrete Mathematical Structures”, Pearson Education, 6th ed., 2008.
4. J. P. Tremblay, R. Manohar, “Discrete Mathematical Structures with Applications to Computer Science”, Tata McGraw Hill, 1st ed., 2001.
5. Susanna S. Epp, “Discrete Mathematics with Applications”, 4th ed., 2010.

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VII	**Course Type: DSE 8**
Course Code:	**Course Title:** Research Methodology	**Credits: L | T | P (3 | 1| 0)**	

Course outcomes:

- Understand the concept of research.
- Understand the concept of data collection and selection for research.
- Understand the applicability of research for public at large.

Course Prerequisites: Basic Knowledge of Computer

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Introduction to Research Methods in science – Philosophy of Science, Research methods and Creative Thinking, Evolutionary Epistemology, Scientific Methods, Hypotheses Generation and Evaluation, Code of Research Ethics, Definition and Objectives of Research, Various Steps in Scientific Research, Research presentations Types of Research – Research Purposes – Research Design – Survey Research – formulation of scientific problems and hypotheses selection of methods for solving a scientific problem Case Study Research.	20
II.	How to perform a literature review – Sampling Methods – Data Processing and Analysis strategies - Data Analysis with Statistical Packages – Statistical Analysis – Hypothesis-testing – Generalization and Interpretation.	20
III.	Research Reports - Structure and Components of Research Report, Types of Report, Layout of Research Report, Mechanism of writing a research report – Requirements of a good dissertation.	20

Textbooks:

1. Oates, B.J., (2005). Researching Information Systems and Computing. Sage Publications, UK.
2. Zobel, J. (2004). Writing for Computer Science - The art of effective communication. 2nd ed., Springer, UK.
3. Garg, B.L., Karadia, R., Agarwal, F. and Agarwal, U.K., 2002. An introduction to Research Methodology, RBSA Publishers.
4. Kothari, C.R., 1990. Research Methodology: Methods and Techniques. New Age International. 5
5. Sinha, S.C. and Dhiman, A.K., 2002. Research Methodology.

Subject: Computer Science

Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VII	**Course Type: DSE 9/GE 8**
Course Code:	**Course Title:** Mobile App Design and Development	**Credits: L | T | P (3 | 0| 1)**	

Course outcomes:

- Learn about mobile devices, Android OS, and architecture, and how to use Android Studio.
- Understand key components like Intents, Manifest files, and permissions.
- Explore activity life cycles, types of intents, and data sharing using intents.
- Develop skills in creating UI elements, handling events, using animations, notifications, and fragments.

Course Prerequisites: GE 4/DSE 2

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	Android Systems: Introduction to Mobile devices and applications, Open Handset Alliance (OHA), Overview of Android OS and architecture, installing android studio. Introduction to Android application components, Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions Activities and intents: understanding activity and its life cycle, Types of intents, intent filter, context, data sharing using intent	12
II.	Android User Interface: Basic android UI, layouts, view and view attributes, button, controls. UI events and event listeners, animations, notifications, progress dialog, Action bar, toolbar, menus and pop ups, Tab based UI, Fragment, Types of Fragment, Fragment Lifecycle, communication between fragment and activity	12
III.	Android Storage and APIs: Android storage: Using Android Data and Storage APIs, Managing data using SQLite, Sharing Data between Applications with Content Providers Android APIs: Multimedia, Using Android Networking APIs, Using Android Web APIs, Using Android Telephony APIs, android location based services	12
IV.	iOS Technology Stack: Introduction to iOS technology stack: iOS architecture, StoryBoard, features of Xcode, components of iOS SDK. Introduction to swift: data types, variables, control flow and operators, Collections and functions in swift, classes and structures, inheritance, closure and. enumerations	12
V.	User interactions: Controls, gesture organizers, touching views, Core Location and Mapkit, using Google Maps in iOS. Sensors in iOS. Data persistence: Core Data framework for storing persistent data, CRUD operations.	12
	Lab: Mobile App Design and Development	15

Textbooks:

1. "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips and Brian Hardy - Offers a comprehensive introduction to Android development.
2. "Android Application Development All-in-One For Dummies" by Barry Burd - Covers various aspects of Android app development, including UI design and application components.
3. "iOS Programming: The Big Nerd Ranch Guide" by Christian Keur and Aaron Hillegass - Provides a detailed guide to iOS development, including Swift programming and user interactions.
4. "Learning Swift: Building Apps for macOS, iOS, and Beyond" by Paris Buttfield-Addison, Jon Manning, and Tim Nugent - Focuses on Swift programming for iOS development.

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VII	**Course Type: Dissertation 1**
Course Code:	**Course Title: Dissertation on Major**	**Credits: 6**	
Course outcomes: • Develop the ability to conduct thorough research, including literature review, problem formulation, and application of appropriate methodologies. • Demonstrate advanced technical skills and proficiency in relevant tools, technologies, and programming languages. • Enhance skills in presenting research findings and defending the dissertation through clear and coherent written and oral communication. • Apply ethical principles in research and cultivate critical thinking skills for evaluating results and making informed conclusions.			
Course Prerequisites: DSE 8			
Course Outline:			
	Writing a sample Research Project/Dissertation/Research Paper on any given topic covering all the components of Research.		

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester: VIII**	**Course Type: DSC 8**
Course Code:	**Course Title:** Compiler Design	**Credits: L | T | P (3 | 1| 0)**	

Course outcomes:

- Understanding of Compiler Architecture
- Ability to Implement Lexical and Syntax Analysis:
- Knowledge of Semantic Analysis and Intermediate Code Generation
- Experience with Compiler Optimization and Code Generation

Course Prerequisites: Basic understanding of compilers and programming languages.

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	Compiler Structure: Analysis-synthesis model of compilation, various phases of a compiler, tool-based approach to compiler construction. Lexical analysis: Interface with input parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, error reporting and implementation. Regular grammar & language definition, Transition diagrams, Design of a typical scanner using LEX or Flex.	12
II.	Syntax Analysis: Context free grammars, ambiguity, associability, precedence, top down parsing, recursive descent parsing, transformation on the grammars, predictive parsing LL(1) grammar, Bottom up parsing, operator precedence grammars, LR parsers (SLR, LALR, LR), Design of a typical parser using YACC or Bison.	12
III.	Syntax directed definitions: Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions. Type checking: type system, type expressions, structural and name equivalence of types, type conversion, overloaded function and operators, polymorphic function. Run time system: storage organization, activation tree, activation record, parameter passing symbol table, dynamic storage allocation. Intermediate code generation: intermediate representation, translation of declarations, assignments, Intermediate Code generation for control flow, Boolean expressions and procedure calls, implementation issues.	12
IV.	Code generation and instruction selection: Issues, basic blocks and flow graphs, register allocation, code generation, DAG representation of programs, code generation from DAGS, peep hole optimization, code generator, specification of machine.	12
V.	Code optimization: Source of optimizations, optimization of basic blocks, loops, global dataflow analysis, solution to iterative dataflow equations, code improving transformations, dealing with aliases, data flow analysis of structured flow graphs.	12

Textbooks:

1. K. C. Louden, "Compiler Construction, Principle and Practice", Cengage Publication, 6th ed., 2009.
2. Alfred V. Aho, Ravi Sethi, Jeffrey, D. Ullman, "Compilers: Priciples, Techniques and Tools", Pearson, 1998.
3. V. Raghvan, "Principles of Compiler Design", TMH, 2009.
4. Levine, Mason, Brown, "Lex & Yacc", O' Reilly, 1998.
5. S. S. Muchnick Harcourt Asra, "Advanced Compiler Design implementation", Morgan Kaufman, 2006.
6. Allen, "Modern Compiler Implementation in C", Cambridge University Press, 1997.
7. Vinu V. Das, "Compiler Design using FLEX and YACC", PHI, 2005.
8. Cooper, "Engineering a Compiler", Elsevier, 2005.
9. Alan I. Holub, "Compiler Design in C", PHI, 2009. 10. Fisher, "Crafting a Compiler in C", Pearson, 2005.
10. Fisher, "Crafting a Compiler in C", Pearson, 2005.

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VIII	**Course Type: DSE 10**
Course Code:	**Course Title:** Design & Analysis of Algorithm	**Credits: L | T | P (3 | 0| 1)**	

Course outcomes:

- Understand concept of asymptotic analysis and perform complexity analysis of iterative and recursive algorithms.
- Formulate and solve time complexity recurrence relations using various techniques.
- Solve computational problems using various algorithmic paradigms like divide-and conquer, greedy, dynamic programming, backtracking, branch-and-bound.

Course Prerequisites: Basic understanding of algorithms and data structures

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Introduction: Review of Asymptotic Notations, Mathematical analysis for Recursive and Non-recursive algorithms, solving recurrence relations.	10
II.	Algorithm Design Techniques: Brute Force, Exhaustive Search, Divide and conquer, Merge sort, Quick sort, Binary search, Multiplication of Large Integers, Strassen's Matrix Multiplication. Greedy strategy– General Approach and problems like Optimal Merge Patterns, Minimum Spanning Trees algorithms, Knapsack Problem, Huffman Code, Job sequencing with deadlines, single source shortest path. Dynamic Programming– General Approach, Memoization, Multistage Graph, Matrix-Chain Multiplication, Longest Common Subsequence, Knapsack Problem, Floyd Warshall algorithm, Optimal Binary Search Trees.	20
III.	Limitations of Algorithm Power: Limitations of Algorithm Power: Lower-Bound Arguments, Decision Trees, P, NP, NP-Hard and NP-Complete Problems, Intractability, Cook's Theorem, Reductions. Coping with the Limitations – Backtracking concept; Branch & Bound method, Approximation Algorithms.	15
	Lab: Design & Analysis of Algorithm	15

Textbooks:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", Prentice Hall of India, 3rd ed., 2010.
2. R. C. T. Lee, S. S. Tseng, R. C. Chang, Y. T. Tsai, "Introduction to the Design and Analysis of Algorithms: A Strategic Approach" McGraw Hill, 2006.
3. Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education,2007.
4. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", UniversityPress, 2nd ed., 2008.
5. Kenneth A. Berman, Jerome Paul, "Algorithms: Sequential, Parallel and Distributed",Cengage Learning, 2004.
6. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.
7. Michael T. Goodrich, Roberto Tamassia, Algorithm Design, Wiley, 2002.
8. S. Dasgupta, C. Papadimitriou, and U. Vazirani. Algorithms. McGraw-Hill Higher Education, 2006

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VIII	**Course Type: DSE 11**
Course Code:	**Course Title:** Artificial Intelligence	**Credits: L | T | P (3 | 1| 0)**	

Course outcomes:

- Understand the concept of Artificial Intelligence.
- Understand to apply the knowledge and reasoning for different components.
- Understand the Expet Systems and thier uses.
- Understand the basics of PROLOG.

Course Prerequisites: DSC 1

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Introduction and applications of artificial intelligence, Problem solving: State space search, Production system, Problem characteristics, Problem system characteristics, Search techniques: Generate and test, Hill climbing, Best first search, A* algorithm, Problem reduction	12
II.	Knowledge and Reasoning: Knowledge acquisition, Knowledge engineer, Cognitive behavior, Knowledge representation: Level of representation, Knowledge representation schemes, Formal logic, Inference Engine, Semantic net, Frame, Scripts. Adversarial search, Optimal and imperfect decisions, Alpha, Beta pruning, Logical agents: Propositional logic, First order logic – Syntax and semantics – Inference in first order logic. Uncertain Knowledge and Reasoning: Uncertainty – Acting under uncertainty – Basic probability notation – Axioms of probability – Baye's rule – Probabilistic reasoning – Making simple decisions.	12
III.	Expert systems: Definition, Role of knowledge in expert system, Architecture, Expert System Development Life Cycle: Problem selection, Prototype construction, Formalization, Implementation, Evaluation	12
IV.	Planning and Learning: Planning: Planning problem – Partial order planning – Planning and acting in non-deterministic domains – Learning: Learning decision trees, Knowledge in learning, Neural networks, Reinforcement learning – Passive and active.	12
V.	PROLOG Programming: Introduction, variables, using rules, Input and Output predicates, Fail and cut predicates, Recursion, Arithmetic operation, Compound object, Dynamic database, Lists, String, File operations.	12

Textbooks:

1. Elaine Rich, Kevin Knight, "Artificial Intelligence", Tata McGraw Hill.
2. Dan W. Patterson, "Introduction to Artificial Intelligence and Expert Systems", Prentice Hall of India.
3. Nils J. Nilsson, "Principles of Artificial Intelligence", Narosa Publication house.
4. Stuart Russell, Peter Norvig, "Artificial Intelligence: A Modern Approach", Pearson Education, 2nded.
5. Winston, Patrick, Henry, "Artificial Intelligence", Pearson Education.
6. Gopal Krishna, Janakiraman, "Artificial Intelligence".

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VIII	**Course Type: DSE 12**
Course Code:	**Course Title:** Computer Graphics	**Credits: L | T | P (3 | 1| 0)**	

Course outcomes:

- Understand the concept of Graphics.
- Understand the concept of Transformation of Images.
- Understand the concept of rendering related to surface.
- Understand the Graphics Programming

Course Prerequisites: DSC 1 & 10+2 with Mathematics

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Introduction: Computer Graphics– Overview, Basic elements, Animation and Multimedia Applications, Pictures– Representation, Storage and Display, Visualization and Image Processing, RGB color model, Output/Display Devices– Cathode Ray Tube (CRT), Refreshing Display Devices– Raster scan display device– Pixel, Frame Buffer, Color Display, Random scan display device, Plotters, Printers, Digitizers, Tablets, Light Pen, 3D viewing devices, Active and Passive Graphic Devices, Software for Computer Graphics. Lines– Point Plotting Techniques, Points and Lines, Line drawing algorithms– Digital Differential Analyzer (DDA) algorithm, Bresenham's algorithm, Circle and Ellipse drawing algorithms, Region filling algorithms– Boundary Seed Fill algorithm, Flood Fill algorithm.	12
II.	Transformations: 2D and 3D Transformations– Translation, Rotation, Scaling and other transformations, Matrix Representation of Points, Homogeneous Coordinate System, 2D and 3D Viewing Transformations, Parallel and Perspective Projections, Clipping and Windowing, Line Clipping algorithms– Cohen-Sutherland Line Clipping algorithm, Cyrus-Beck Line Clipping algorithm.	12
III.	Curves and Surfaces: Curve representation, Polygon representation methods, Bezier curves, Bezier surfaces, Spline representations, B-spline methods, Hidden Surface Removal– Z-buffer algorithm, Back face detection, Binary Space Partitioning (BSP) tree method, Scan Line Coherence algorithm, Hidden Line Elimination	12
IV.	Surface Rendering: Illumination/Lighting and Shading Models, Surface Lighting Effects, Basic Lighting Models– Ambient Lighting, Diffuse Lighting, Specular Reflection Lighting Model (Phong Specular Reflection Model), combined effect of Ambient, Diffuse and/or Specular Reflection. Shading– Gouraud Shading, Phong Shading Model, Creating Shadowed Objects, Drawing Shadows, Rendering Texture.	12
V.	Graphics Programming: Graphics Programming using OpenGL, Programming 2D Applications, The OpenGL API, Primitives and Attributes, Color, Viewing, Control Functions, Polygons and Recursion, The Three-Dimensional Gasket, Plotting Implicit Functions, Interaction, Input Devices, Clients and Servers, Display Lists.	12

Textbooks:

1. F. S. Hill, "Computer Graphics Using OpenGL", Pearson Education, 2nded., 2007.
2. Donald D. Hearn, M. Pauline Baker, "Computer Graphics with OpenGL", Pearson Education, 3rded., 2004.
3. David Rogers, "Procedural Elements of Computer Graphics", McGraw Hill, 2nded., 2001.

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VIII	**Course Type: DSE 13**
Course Code:	**Course Title:** Web App Design and Development		**Credits: L | T | P (3 | 0| 1)**

Course outcomes:

- Develop interfaces for single page applications
- Develop a complete client-side solutions using angular js
- Develop a RESTful web service.
- Apply form validations

Course Prerequisites: DSC 1 & DSC 5

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Introduction to React: Definition of React, React library, React Developer tools, Introduction to ES6, Declaring variables, Arrow Functions, Objects and Arrays, modules, Introduction to AJAX, Functions in AJAX Pure React: Page setup, virtual DOM, React Element, React DOM, Constructing Elements with Data, React Components, DOM Rendering, First React Application using Create React App, React with JSX, React Element as JSX Props, State and Component Tree: Property Validation, Validating Props with createClass, Default Props, ES6 Classes and stateless functional components, React state management, State within the component tree, state vs props, Forms in React	20
II.	Rest APIs: JSON: Introduction, Syntax, Data Types, Objects, Schema. REST API: Introduction, WRML, REST API Design, Identifier Design with URIs, Interaction Design with HTTP, Representation Design, Caching, Security.	10
III.	Angular.js.: Introduction to Angular: Angular architecture; introduction to components, component interaction and styles; templates, interpolation and directives; forms, user input, form validations; data binding and pipes; retrieving data using HTTP; Angular modules	15
	Lab: Web App Design and Development	15

Textbooks:

1. D. Brad, B. Dayley and C. Dayley, Node. js, MongoDB and Angularjs Web Development: The definitive guide to using the MEAN stack to build web applications, 2nd edition, Addison-Wesley, 2018.
2. D. Herron, Node.js Web Development, 5thedition, Packt Publishing, 2020.
3. A. Banks and E. Porcello, Learning React: Functional Web Development with React and Redux, 1st edition, O'Reilly, 2017.
4. M. Masse, REST API – Design Rulebook, 1st edition, O'Reilly, 2011.

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VIII	**Course Type: DSE 14/GE 9**
Course Code:	**Course Title:** Cloud Computing		**Credits: L | T | P (3 | 1| 0)**

Course outcomes:

- Understand the evolution of Cloud Computing and compare with traditional Computing
- Remember the key terminologies used in Cloud Computing and understand key concepts
- Describe virtualization architecture and implement the virtualization using open-source tools
- Identify the advantages and disadvantages of various cloud computing platforms and service models.
- Classify security and privacy issues in cloud computing.
- Apply various cloud services to understand elasticity, scalability and availability properties of Cloud services and also their usage towards web service deployments.

Course Prerequisites: DSC 6

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Overview of Computing Paradigm: History with overview of Computing Paradigm, Cluster Computing, Grid Computing, Distributed Computing, Utility Computing, Cloud Computing versus Traditional Computing	8
II.	Introduction to Cloud Computing: Introduction to Cloud Computing, Different Perspectives on Cloud Computing, Characteristics, Different Stakeholders in Cloud Computing, Cloud NIST Reference Architecture	8
III.	Service Level Agreements (SLAs), Total cost of ownership (TCO), Benefits and limitations of Cloud Computing, Open Challenges	8
IV.	Virtualization: Introduction & need of Virtualization, Definition & types of Hypervisors, Characteristics of Virtualized Environments, Virtualization and Cloud Computing, System calls & Ring Privileges, Machine Reference Architecture, Xen Hypervisor Architecture, Pros and Cons of Virtualization	8
V.	Cloud Computing Architecture: Traditional Computing Architecture-Client-Server Architecture, Peer to Peer Architecture, OpenStack-based Cloud Computing Architecture, Cloud Reference Architecture: Service Models Perspective-Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Deployment Models- Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud	10
VI.	Cloud Security: Introduction, Cloud Security Issues such as Application-level Security, Network Level Security, Data-level Security, Virtualization Security, Identity Management & Access Control	9
VII.	Case Studies: Implementation of Cloud Services: AWS Cloud Services, Google Cloud Services, Apply Cloud Services for Hosting the Website	9

Textbooks:

1. Raj Kumar Buyya, Mastering the Cloud Computing,MacGraw Hill Education (India), 2013 2. Tim Mather, SubraKumaraswamy, ShahedLatif: Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance 3. J.R. ("Vic") Winkler: Securing the Cloud

4. Haley Beard, Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing, Applications and Data Centers in the Cloud with SLAs, Emereo Pty Limited, July 2008. Reference Books:

Reference Books:

1. Michael Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Que Publishing, August 2008.

2. David Chisnall, The Definitive Guide to Xen Hypervisor, Prentice Hall; Reprint edition (9 November 2007)

Subject: Computer Science			
Programme/Class: B.Sc. (H)	**Year:** 4th	**Semester:** VIII	**Course Type: Dissertation 2**
Course Code:	**Course Title: Dissertation on Major**		**Credits: 6**

Course outcomes:

- Develop the ability to conduct thorough research, including literature review, problem formulation, and application of appropriate methodologies.
- Demonstrate advanced technical skills and proficiency in relevant tools, technologies, and programming languages.
- Enhance skills in presenting research findings and defending the dissertation through clear and coherent written and oral communication.
- Apply ethical principles in research and cultivate critical thinking skills for evaluating results and making informed conclusions.

Course Prerequisites: DSE 8

Course Outline:

	Writing a sample Research Project/Dissertation/Research Paper on any given topic covering all the components of Research.	

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** IX	**Course Type: DSC 9**
Course Code:	**Course Title:** Machine Learning		**Credits: L | T | P (3 | 0| 1)**

Course outcomes:

- Understanding popular ML algorithms with their associated mathematical foundations for appreciating these algorithms.
- Capability to implement basic algorithms using basic machine learning libraries mostly in python. Gain hands-on experience in applying ML to problems encountered in various domains. In addition, obtain exposure to high-level ML libraries or frameworks such as TensorFlow, PyTorch.
- Make aware of the role of data in the future of computing, and also in solving real-world problems using machine learning algorithms.
- Help connect real-world problems to appropriate ML algorithm(s) for solving them. Enable formulating real world problems as machine learning tasks.
- Appreciate the mathematical background behind popular ML algorithms.
- Ensure awareness about importance of core CS principles such as algorithmic thinking and systems design in ML

Course Prerequisites: GE 6/DSE 4 & GE 7/DSE 5, Fundamental statistical concepts and probability theory & 10+2 Mathematics

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	Introduction to ML: (Motivation and role of machine learning in computer science and problem solving. Representation (features), linear transformations, Appreciate linear transformations and matrix vector operations in the context of data and representation. Problem formulations (classification and regression). Appreciate the probability distributions in the context of data, Prior probabilities and Bayes Rule. Introduce paradigms of Learning (primarily supervised and unsupervised. Also, a brief overview of others)	11
II.	Fundamentals of ML: PCA and Dimensionality Reduction, Nearest Neighbours and KNN. Linear Regression, Decision Tree Classifiers, Notion of Generalization and concern of Overfitting, Notion of Training, Validation and Testing; Connect to generalisation and overfitting.	11
III.	Selected Algorithms: Ensembling and RF, Linear SVM, K Means, Logistic Regression, Naive Bayes	11
IV.	Neural Network Learning: Role of Loss Functions and Optimization, Gradient Descent and Perceptron/Delta Learning, MLP, Backpropagation, MLP for Classification and Regression, Regularisation, Early Stopping, Introduction to Deep Learning, CNNs	12
	Lab: Machine Learning	15

Textbooks:

1. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Mathematics for Machine Learning, Cambridge University Press (23 April 2020)
2. Tom M. Mitchell- Machine Learning- McGraw Hill Education, International Edition
3. Aurélien Géron Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, Inc. 2nd Edition

Reference Books:

1. Ian Goodfellow, Yoshoua Bengio, and Aaron Courville Deep Learning MIT Press Ltd, Illustrated edition
2. Christopher M. Bishop Pattern Recognition and Machine Learning- Springer, 2nd edition
3. Trevor Hastie, Robert Tibshirani, and Jerome Friedman - The Elements of Statistical Learning: Data Mining, Inference, and Prediction- Springer, 2nd edition

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** IX	**Course Type: DSE 15**
Course Code:	**Course Title:** Data Mining & Warehousing	**Credits: L | T | P (3 | 1| 0)**	

Course outcomes:

- Understand the Data Mining basics and knowledge Discovery in Database.
- Understand the pattern identification and knowledge recognition.
- Understand the basics of classification, clustering and their related techniques. .

Course Prerequisites: GE 6/DSE 4, Fundamental statistical concepts and probability theory & 10+2 Mathematics

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Evolution of database technology: Introduction to data warehousing and data mining, difference between operational databases and data warehouses.	12
II.	Data warehouse architecture & design: Data warehousing Components, building a Data warehouse, Mapping the Data warehouse to multiprocessor architecture, DBMS Schema as for Decision Support, Data Extraction, Clean up and Transformation tools, Metadata.	12
III.	Data mining: Data Pre-processing & Data Mining Primitives Data Pre-processing, Data cleaning, Data Integration and Transformation, Data reduction, Discretization and Concept Hierarchy Generation, Data Mining primitives, Languages and System Architectures, Concept Description: characterization and Comparison, Analytical Characterization, Mining Class Comparison.	12
IV.	Association Rules & Mining Association Rule Mining: Mining of Single dimensional Boolean association rules, Constraint based association Mining Classification and prediction: Basic issues regarding classification and prediction, Classification by Decision Tree, Bayesian classification, Prediction, Classifier accuracy.	12
V.	Cluster Analysis: Basic issues, clustering using partitioning methods, Hierarchical methods, Density based methods, Grid based methods and model-based methods, Algorithms for outlier analysis.	12

Textbooks:

1. Ralph Kimball, “The Data Warehouse Life Cycle Toolkit”, John Wiley & Sons Inc., 1998.
2. Alex Berson, S.J. Smith, “Data Warehousing, Data Mining & OLAP”, TMH, 1997.
3. W.H. Inmon, “Building the Data Warehouse”, Wiley India, 2011.

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** IX	**Course Type: DSE 16**
Course Code:	**Course Title:** Cryptography & Network Security		**Credits: L | T | P (3 | 1| 0)**

Course outcomes:

- Understand the evolution of Cryptography.
- Understand the different type of Authentication and messages.
- Understand the basics of web and system security.

Course Prerequisites: computer science fundamentals, networking basics, mathematics (especially number theory, algebra, and probability theory), operating systems knowledge, security basics, programming skills, database knowledge, web technologies familiarity, discrete mathematics understanding, and Linux/Unix basics.

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Introduction to cryptography: Private key cryptography, Conventional Encryption models, Classical encryption techniques, Substitution cipher, Transposition cipher, Cryptanalysis, Stereography, Stream and block ciphers, Modern block cipher: principles, Shannon's theory of confusion and diffusion, Fiestal structure, DES, Strength of DES, Triple DES, AES, IDEA, Key distribution, Diffie-Hellman algorithm, Public key cryptography, RSA algorithm, Elliptic curve cryptography, Elgamal cryptosystem.	15
II.	Message Authentication and Hashing: Authentication requirements, Message Digest Algorithms-MD4, MD5, Hash functions, Security of hash functions, Message Authentication Codes (MAC), Secure hash algorithm (SHA). Digital Signatures: Digital Signatures, Authentication protocols, Digital signature standards.	15
III.	Message Authentication and Hashing: Authentication requirements, Message Digest Algorithms-MD4, MD5, Hash functions, Security of hash functions, Message Authentication Codes (MAC), Secure hash algorithm (SHA). Digital Signatures: Digital Signatures, Authentication protocols, Digital signature standards.	15
IV.	Web and System Security: Secure socket layer (SSL), Transport layer security, Secure electronic transaction (SET). System Security: Intruders, Intrusion Detection System, Password Management, Viruses and related threats, Distributed Denial of Service Attacks, Firewalls, Firewall design principles, Trusted systems.	15

Textbooks:

1. William Stallings, Cryptography and Network Security: Principals and Practice, Pearson Education, 6thed., 2013.
2. B. Forouzan, Cryptography and Network Security, TMH, 2nded., 2010.
3. AtulKahate, Cryptography and Network Security, TMH, 7thed., 2013.
4. Johannes A. Buchmann, Introduction to Cryptography, Springer, 2nded., 2009.
5. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** IX	**Course Type: DSE 17**
Course Code:	**Course Title:** Quantum Computing	**Credits: L | T | P (3 | 1| 0)**	

Course outcomes:

- Understand basics of quantum computing
- Understand physical implementation of Qubit
- Understand Quantum algorithms and their implementation
- Understand The Impact of Quantum Computing on Cryptography

Course Prerequisites: Linear Algebra, Prior knowledge of quantum mechanics.

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Introduction to Essential Linear Algebra: Some Basic Algebra, Matrix Math, Vectors and Vector Spaces, Set Theory. Complex Numbers: Definition of Complex Numbers, Algebra of Complex Numbers, Complex Numbers Graphically, Vector Representations of Complex Numbers, Pauli Matrice, Transcendental Numbers.	12
II.	Basic Physics for Quantum Computing: The Journey to Quantum, Quantum Physics Essentials, Basic Atomic Structure, Hilbert Spaces, Uncertainty, Quantum States, Entanglement. Basic Quantum Theory: Further with Quantum Mechanics, Quantum Decoherence, Quantum Electrodynamics, Quantum Chromodynamics, Feynman Diagram Quantum Entanglement and QKD, Quantum Entanglement, Interpretation, QKE.	12
III.	Quantum Architecture: Further with Qubits, Quantum Gates, More with Gates, Quantum Circuits, The D-Wave Quantum Architecture. Quantum Hardware: Qubits, How Many Qubits Are Needed? Addressing Decoherence, Topological Quantum Computing, Quantum Essentials	12
IV.	Quantum Algorithms: What Is an Algorithm? Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Bernstein-Vazirani Algorithm, Simon's Algorithm, Shor's Algorithm, Grover's Algorithm.	12
V.	Current Asymmetric Algorithms: RSA, Diffie-Hellman, Elliptic Curve The Impact of Quantum Computing on Cryptography: Asymmetric Cryptography, Specific Algorithms, Specific Applications.	12

Textbooks:

1. Quantum Computing for Computer Scientists by Noson S. Yanofsky and Mirco A. Mannucci.
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. Basic Concepts, Vol
3. Basic Tools and Special Topics, World Scientific. Pittenger A. O., An Introduction to Quantum Computing Algorithms

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** IX	**Course Type:** DSE 18/GE 10
Course Code:	**Course Title:** Internet of Things	**Credits: L | T | P (2 | 0| 2)**	

Course outcomes:

- Design and outline IoT architectures, considering design principles and standards.
- Understand devices, gateways, and networking concepts in IoT and M2M communications.
- Learn about technical design constraints, data representation, visualization, and remote control in IoT systems.
- Gain knowledge of IoT data link, network, transport, and session layer protocols, including their functionalities and use cases.

Course Prerequisites: programming skills (Python, C, Java), networking knowledge (TCP/IP, HTTP), electronics understanding, data analytics, security principles, cloud computing familiarity, machine learning basics, embedded systems, and knowledge of IoT platforms and communication protocols.

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	IoT-An Architectural Overview– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, and standards considerations. M2M and IoT Technology Fundamentals- Devices and gateways, Local and wide area networking	6
II.	Reference Model and architecture, IoT reference Model - IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. Real-World Design Constraints- Introduction, Technical Design constraints- hardware is popular again, Data representation and visualization, Interaction, and remote control	6
III.	IOT DATA LINK LAYER & NETWORK LAYER PROTOCOLS: PHY/MAC Layer (3GPP MTC, IEEE 802.11, IEEE 802.15), Wireless HART, Z Wave, Bluetooth Low Energy, Zigbee Smart Energy, DASH7 - Network Layer-IPv4, IPv6, 6LoWPAN, 6TiSCH, ND, DHCP, ICMP, RPL, CORPL, CARP	6
IV.	TRANSPORT & SESSION LAYER PROTOCOLS: Transport Layer (TCP, MPTCP, UDP, DCCP, SCTP)- (TLS, DTLS) – Session Layer HTTP, CoAP, XMPP, AMQP, MQTT	6
V.	SERVICE LAYER PROTOCOLS & SECURITY: Service Layer -oneM2M, ETSI M2M, OMA, BBF – Security in IoT Protocols – MAC 802.15.4, 6LoWPAN, RPL, Application Layer	6
	Lab: Internet of Things	30

Textbooks:

1. "Architecting the Internet of Things" by Dieter Uckelmann, Mark Harrison, and Florian Michahelles - Offers insights into IoT architectures and design principles.
2. "Internet of Things (IoT): Technologies, Applications and Implementations" by B. S. Chandra Sekhar - Provides a comprehensive overview of IoT technologies and applications.
3. "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things" by David Hanes and Gonzalo Salgueiro - Covers IoT networking technologies and protocols.
4. "Internet of Things: Principles and Paradigms" by Rajkumar Buyya, Amir Vahid Dastjerdi, and Satish Narayana Srirama - Discusses IoT concepts, architecture, and protocols in detail.

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** IX	**Course Type: Dissertation 3**
Course Code:	**Course Title: Dissertation on Major**	**Credits: 6**	
Course outcomes: • Develop the ability to conduct thorough research, including literature review, problem formulation, and application of appropriate methodologies. • Demonstrate advanced technical skills and proficiency in relevant tools, technologies, and programming languages. • Enhance skills in presenting research findings and defending the dissertation through clear and coherent written and oral communication. • Apply ethical principles in research and cultivate critical thinking skills for evaluating results and making informed conclusions.			
Course Prerequisites: DSE 8			
Course Outline:			
	Writing a sample Research Project/Dissertation/Research Paper on any given topic covering all the components of Research.		

Subject: Computer Science

Programme/Class: M.Sc. **Year:** 5th **Semester:** X **Course Type: DSC 10**

Course Code: **Course Title:** Digital Image Processing **Credits: L | T | P (3 | 0| 1)**

Course outcomes:

- Review the fundamental concepts of a digital image processing system.
- Analyze images in the frequency domain using various transforms
- Evaluate the techniques for image segmentation and object detection.
- Categorize various compression techniques.

Course Prerequisites: Foundational knowledge in mathematics, particularly in linear algebra and calculus, and basic programming skills.

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Digital Image Processing System, Image Perception, Colour Representation, Image Acquisition, Image Digitization, Image model, Image scanning techniques, Noise, Image Processors, A brief overview of OpenCV, Installing OpenCV in Windows, Linux, how are Images formed and stored	9
II.	Gray Level Transformation, Histogram Processing, Grey Level Transformation Techniques, Multi Image Operations, OpenCV-Gray scaling, histogram representation of images, drawing over images, Transformation, Scaling, Cropping, Darkening/Brightening Images, Masking Blurring, and Sharpening	9
III.	Segmentation, Region Based Segmentation, Thresholding, Basic Edge Detection, Colour Edge Detection, Pyramid Edge Detection, OpenCV – Edge Detection using Image Gradient, Segmentation and Contours, Link Detection, Circle Detection, and Blob Detection	9
IV.	System Component, Complexity of Object Recognition, Object Representation, Feature Detection, Recognition Strategies. OpenCV-Finding Corners, Extracting Features, Face Detection using HAAR's Cascade. Basic morphology operations: dilation and erosion, Structuring elements and their properties, Opening and closing operations, Hit-or-miss transform, Boundary extraction	9
V.	Introduction to Image compression, Basic Requirements, Different Types of Compressions, Coding algorithms: Run Length Coding, Huffman Coding LZW, JPEG	9
	Lab: Digital Image Processing	15

Textbooks:

1. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, “Digital Image Processing using MATLAB”, PHI, 2003.
2. Anil K. Jain, “Fundamentals of Digital Image Processing”, Prentice Hall, 1989.
3. Digital Image Processing, Rafael C. González, Richard Eugene Woods, Steven L., Pearson, 2010.

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** X	**Course Type: DSE 19**
Course Code:	**Course Title:** Natural Language Processing	**Credits: L | T | P (3 | 0| 1)**	

Course outcomes:

- Learn basics of speech and text processing.
- Understand sequential modeling and algorithms.
- Understand parsing and ambiguity resolution.
- Understand multilinguality and associated applications.

Course Prerequisites: Familiarity with Python programming and basic understanding of linguistics and machine learning.

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	Biology of Speech Processing; Place and Manner of Articulation; Word Boundary Detection; Argmax based computations; HMM and Speech Recognition.	9
II.	Words and Word Forms: Morphology fundamentals; Morphological Diversity of Indian Languages; Morphology Paradigms; Finite State Machine Based Morphology; Automatic Morphology Learning; Shallow Parsing; Named Entities; Maximum Entropy Models; Random Fields.	9
III.	Theories of Parsing, Parsing Algorithms; Constituency Parsing, Dependency Parsing, Robust and Scalable Parsing on Noisy Text as in Web documents; Hybrid of Rule Based and Probabilistic Parsing; Scope Ambiguity and Attachment Ambiguity resolution.	9
IV.	Lexical Knowledge Networks, Wordnet Theory; Indian Language Wordnets and Multilingual Dictionaries; Semantic Roles; Word Sense Disambiguation; WSD and Multilinguality; Metaphors; Coreferences.	9
V.	Text summarization, Text classification, Sentiment Analysis; Text Entailment; Robust and Scalable Machine Translation; Question Answering in Multilingual Setting; Cross Lingual Information Retrieval (CLIR).	9
	Lab: Natural Language Processing	15

Textbooks:

1. James Allen, “Natural Language Understanding”, Pearson Education, 2nded., 2003.
2. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.
3. C. Manning and S. Heinrich, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
4. Radford, Andrew et. al., Linguistics: An Introduction, Cambridge University Press, 1999.
5. L.M. Ivansca, S.C. Shapiro, “Natural Language Processing and Language Representation”.
6. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008.
7. T. Winograd, “Language as a Cognitive Process”, Addison-Wesley.

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** X	**Course Type: DSE 20**
Course Code:	**Course Title:** Advance Java	**Credits: L | T | P (3 | 0| 1)**	

Course outcomes:

- Demonstrate a solid understanding of core Java concepts, including arrays, strings, and multithreading.
- Apply Java frameworks such as Servlets, JSP, and EJB to develop robust web applications.
- Integrate Java applications with databases using JDBC for efficient data management.
- Develop and consume SOAP and RESTful web services in Java, using XML, JSON, WSDL, and UDDI.

Course Prerequisites: Proficiency in core Java programming, including object-oriented principles and basic Java concepts.

Course Outline:

Units	**Topics**	**No. of Lab/Lectures**
I.	Array and String, Multithreading, Collection Framework, Java Generics, Java Database Connectivity (JDBC) Java Server Pages (JSP): Introduction to JSP, JavaBeans, JSP tags, Expression Language (EL)	9
II.	Servlets: Introduction to Servlets, HTTP Protocol, Servlet Life Cycle, Servlet API Enterprise Java Beans (EJB): Introduction to EJB, Session Beans, Entity Beans, Message-Driven Beans	9
III.	Web Services: Introduction to Web Services, SOAP and RESTful web services, XML and JSON, WSDL and UDDI Design Patterns: Introduction to Design Patterns, Creational patterns, Structural patterns, Behavioral patterns	9
IV.	Spring Framework: Introduction to Spring Framework, Inversion of Control (IoC), Dependency Injection (DI), Spring MVC framework	9
V.	Hibernate: Introduction to Hibernate, Hibernate architecture, Hibernate mapping, Hibernate Query Language (HQL)	9
	Lab: Advance Java	15

Textbooks:

1. "Head First Java" by Kathy Sierra and Bert Bates.
2. "Java: The Complete Reference" by Herbert Schildt.
3. "Core Servlets and JavaServer Pages" by Marty Hall.

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** X	**Course Type: DSE 21**
Course Code:	**Course Title:** System of Cyber Security	**Credits: L | T | P (3 | 0| 1)**	

Course outcomes:

- Understand Linux history, installation, directory structure, basic commands, and permissions.
- Learn about I/O redirection, compression, backup, disk recovery, file processing, and system logs.
- Configure SSH, DNS, web, FTP, and database servers.
- Study system hacking processes, prevention methods, malware threats, and analysis techniques.

Course Prerequisites: Basic understanding of computer networks and operating systems.

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	Linux Basics: Introduction to Linux, History of Unix and Linux, Installation of Kali Linux, Directory Structure, Basic Commands, VI editor, Permissions, User and Groups.	9
II.	Advanced Linux: I/O redirectors, Hardlink and Softlink, Compression/Decompression, Backup and Scheduling Tasks, Disk Recovery using Foremost and ddrescue, Filter Commands, Finding and processing Files, Process Commands, analysing logs, Exploring Virtual File System (Proc)	9
III.	Managing Services using Linux: Configuring SSH Server, Configuring DNS Server, Configuring Web Server with Virtual Hosting, Configuring FTP Server, Configuring Database Server (MySQL).	9
IV.	System Hacking Process: CEH System Hacking Process, hacking tools – keyloggers, spywares and rootkits, etc., System Hacking Prevention, Penetration testing Steps.	9
V.	Malware Threats: Malware Introduction, Malware Propagation Techniques, Working of Trojans and Viruses, Static and Dynamic Malware analysis process, Methods of Virus Detection	9
	Lab: System of Cyber Security	15

Textbooks:

1. "Linux Bible" by Christopher Negus - Provides comprehensive coverage of Linux basics and advanced topics.
2. "Linux Command Line and Shell Scripting Bible" by Richard Blum and Christine Bresnahan - Offers in-depth guidance on Linux command-line usage and scripting.
3. "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto - Focuses on web application security, including hacking methodologies and countermeasures.
4. "Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code" by Michael Ligh, Steven Adair, Blake Hartstein, and Matthew Richard - Covers malware analysis techniques and tools.

Subject: Computer Science			
Programme/Class: M.Sc.	**Year:** 5th	**Semester:** X	**Course Type: DSE 22/GE 11**
Course Code:	**Course Title:** Web Hacking and Security	**Credits: L | T | P (3 | 0| 1)**	

Course outcomes:

- Understand DoS/DDoS attacks, botnet types, and attack tools.
- Detect, prevent, and mitigate DoS/DDoS attacks.
- Recognize and prevent session hijacking at network and application levels.
- Identify attack methods, detect hacking attempts, and implement defenses for web servers and applications.

Course Prerequisites: Basic knowledge of web technologies and programming.

Course Outline:

Units	Topics	No. of Lab/Lectures
I.	DDos Attacks and Prevention: This module explains DoS/DDoS attacks, the classification of DoS/DDoS attacks, and various attack techniques, Discusses Botnets, the types of bots, and how they infect the system, demonstrates various tools to perform DoS and DDoS attacks, discusses various techniques to detect, prevent, and mitigate DoS/DDoS attacks, Briefs about various post-attack forensic methods	9
II.	Session Hijacking: Session hijacking concepts, discusses about network and application-level session hijacking, explains various session hijacking tools, explains various session hijacking detection methods and tool, explains countermeasures to prevent session hijacking attacks	9
III.	Evading IDS, Firewalls and Honeypots: Introduction to IDS, firewall and honeypot concepts and types, demonstrates various IDS, firewall and honeypot solutions, describes various IDS and firewall evasion techniques, explains various techniques to detect and defeat honeypots, lists various IDS/firewall evasion tools and honeypot detection tools, Discusses the countermeasures to defend against IDS/firewall evasion	9
IV.	Hacking Web Servers: Open-source web server and IIS architecture, discusses various reasons why web servers are compromised, demonstrates various key web server attack techniques and tools, discusses about web server attack methodology and tools, discusses various methods to detect web server hacking attempts, explains countermeasures to prevent web server attacks	9
V.	Hacking Web Application: Lists and explains various web application threats and attacks, explains web application hacking methodology, demonstrates various web application hacking tools, SQL Injection Discusses countermeasures to defend against web application attacks, Demonstrates various web application security tools	9
	Lab: Web Hacking and Security	15

Textbooks:

1. "Hacking: The Art of Exploitation" by Jon Erickson - Provides an in-depth understanding of hacking techniques and methodologies, including DoS/DDoS attacks.
2. "Network Security Essentials: Applications and Standards" by William Stallings - Covers topics related to network security, including session hijacking and IDS/Firewall evasion.
3. "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto - Focuses on web application security, including hacking methodologies and countermeasures.
4. "Firewalls and Internet Security: Repelling the Wily Hacker" by William R. Cheswick, Steven M. Bellovin, and Aviel D. Rubin - Provides insights into firewall technologies and evasion techniques.

<table>
<tr><td colspan="4"><b>Subject:</b> Computer Science</td></tr>
<tr><td><b>Programme/Class: M.Sc.</b></td><td><b>Year:</b> 5th</td><td><b>Semester:</b> X</td><td><b>Course Type: Dissertation 4</b></td></tr>
<tr><td><b>Course Code:</b></td><td colspan="2"><b>Course Title: Dissertation on Major</b></td><td><b>Credits: 6</b></td></tr>
<tr><td colspan="4"><b>Course outcomes:</b>
• Develop the ability to conduct thorough research, including literature review, problem formulation, and application of appropriate methodologies.
• Demonstrate advanced technical skills and proficiency in relevant tools, technologies, and programming languages.
• Enhance skills in presenting research findings and defending the dissertation through clear and coherent written and oral communication.
• Apply ethical principles in research and cultivate critical thinking skills for evaluating results and making informed conclusions.</td></tr>
<tr><td colspan="4"><b>Course Prerequisites: DSE 8, Dissertation 3</b></td></tr>
<tr><td colspan="4"><b>Course Outline:</b></td></tr>
<tr><td></td><td colspan="2">Writing a sample Research Project/Dissertation/Research Paper on any given topic covering all the components of Research.</td><td></td></tr>
</table>