

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	Mobile App Design and Development	×
		Discrete Mathematics		
		Research Methodology		
		Mobile App Design and Development		
VIII	Compiler Design	Design & Analysis of Algorithm	Cloud Computing	×
		Artificial Intelligence		
		Computer Graphics		
		Cloud Computing		
		Web App Design and Development		
IX	Machine Learning	Data Mining & Warehousing	Internet of Things	×
		Cryptography & Network Security		
		Internet of Things		
		Quantum Computing		
X	Digital Image Processing	Natural Language Processing	Web Hacking and Security	×
		Advance Java		
		Web Hacking and Security		

## Definition of Credit

1 Hr. Lecture (L) per week	<b>1 Credit</b>
1 Hr. Tutorial (T) per week	<b>1 Credit</b>
1 Hr. Practical (P) per week	<b>0.5 Credit</b>
2 Hours Practical (P) per week	<b>1 Credit</b>

### 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):	
<b>PO 1</b>	Gain a complete exposure to the theories and practices of Computer science.
<b>PO 2</b>	Get transformed into a skilled learner and active programmer, enabling the students to focus on their higher studies.
<b>PO 3</b>	Value computer professionals and programmers.
<b>PO 4</b>	Explore how the concepts and applications of Computer science led to innovative thinking with a problem-solving attitude.
Programme specific outcomes (PSOs): Certificate Course	
<b>PSO 1</b>	Develop a strong foundation in computer fundamentals, including hardware, software, and operating systems, enabling students to understand and operate various computing environments effectively.
<b>PSO 2</b>	Cultivate proficiency in C++ programming and problem-solving techniques, allowing students to design, implement, and test algorithms to solve real-world problems.
<b>PSO 3</b>	Gain expertise in data structures, enhancing the ability to organize, manage, and analyze data efficiently for optimized performance in computational tasks.
<b>PSO 4</b>	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.
<b>PSO 5</b>	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	
<b>PSO 1</b>	Understand and design digital systems and computer architectures, providing a solid foundation in the principles and practices of hardware and system-level design.
<b>PSO 2</b>	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.
<b>PSO 3</b>	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.
<b>PSO 4</b>	Master the use of OpenOffice Impress for creating and delivering effective presentations, enhancing communication and presentation skills for academic and professional settings.
<b>PSO 5</b>	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)	
<b>PSO 1</b>	Develop a comprehensive understanding of database management systems, including database design, implementation, and administration, enabling students to efficiently manage and manipulate data.
<b>PSO 2</b>	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.
<b>PSO 3</b>	Acquire expertise in client-side and server-side web technologies, allowing students to design, develop, and deploy dynamic and responsive web applications.
<b>PSO 4</b>	Understand the fundamental concepts and functionalities of operating systems, including process management, memory management, and file systems, ensuring effective system-level programming and administration.
<b>PSO 5</b>	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)**

<b>PSO 1</b>	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.
<b>PSO 2</b>	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.
<b>PSO 3</b>	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.
<b>PSO 4</b>	Master the design and analysis of algorithms and artificial intelligence techniques, allowing students to develop optimized solutions and intelligent systems for real-world applications.
<b>PSO 5</b>	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.
<b>PSO 6</b>	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)**

<b>PSO 1</b>	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.
<b>PSO 2</b>	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.
<b>PSO 3</b>	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.
<b>PSO 4</b>	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.
<b>PSO 5</b>	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.
<b>PSO 6</b>	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: 1 <sup>st</sup>	Semester: I	Course Type: DSC 1
Course Code:	Course Title: Computer Fundamentals & Problem-Solving using C++		Credits: L   T   P (3   0   1)
<b>Course outcomes:</b> <ul style="list-style-type: none"> <li>• Bridge the fundamental concepts of computers with the present level of knowledge of the students.</li> <li>• Familiarize operating systems, programming languages, peripheral devices, networking, multimedia and internet</li> <li>• Understand binary, hexadecimal and octal number systems and their arithmetic.</li> <li>• Understand the difference between the top-down and bottom-up approach and concepts of object-oriented programming in connection with C++.</li> <li>• Illustrate the process of data file manipulations using C++ and apply virtual and pure virtual function &amp; complex programming situations.</li> </ul>			
<b>Course Prerequisites:</b> Basic understanding of computer fundamentals and logic.			
<b>Course Outline:</b>			
Unit	Topics		No. of Lab/Lectures
I.	<b>Introduction to Computer:</b> 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.	<b>Memory:</b> Memory hierarchy, Registers (Types of Registers), Cache Memory. <b>Primary Memory</b> (RAM, how data is stored in a RAM, DRAM and SRAM. ROM (BIOS/Firmware & Types of ROM). <b>Secondary Memory</b> (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.	<b>Software:</b> Software and its Need, Types of Software: - System software, Application software. <b>Operating System:</b> History of Operating System, Function of Operating System, OS classification (Batch, Multiprogramming, Multitasking, Multithreading, Multiprocessing, Multiuser, Time sharing, real time). Programming languages, <b>Translators:</b> Compiler, Interpreter and Assembler. <b>Network Fundamental:</b> Categories, Data flow, Topology.		6
IV.	<b>Fundamentals of C++:</b> Data Types and Sizes, Declaration of variables, Modifiers, Identifiers and keywords, Symbolic constants. Operators, Precedence and order of evaluation. <b>Control statements:</b> if-else, else-if clause, switch. Loops: for, while, do-while, break, continue. <b>Functions:</b> Defining a function, function prototyping and function calls, function arguments, passing by reference, inline functions, and default arguments. <b>Arrays:</b> linear arrays, multidimensional arrays, passing arrays to functions.		6
V.	<b>Object Oriented Concepts:</b> Elements of Object-Oriented programming, Objects, Classes, and OOPs features. <b>Classes &amp; Objects:</b> Specifying a Class, Creating Objects, Accessing Class members, defining member function, Outside Member Functions as inline, Accessing Member Functions within		6

	the class, Static data member, Access Specifiers, Constructors and Destructors, Exception Handling basics	
VI.	<b>Operator Overloading:</b> Definition, Overloadable Operators, Unary and Binary Operators overloading through Member Functions and Friend Functions, Function Overloading, Constructor Overloading. <b>Dynamic Memory Allocation:</b> Pointers to Objects, Creating and Deleting Dynamic Objects: New and Delete operators, Array of Objects, Array of Pointers to Objects, Pointers to Object Members, 'this' Pointer.	7
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
<b>Lab: Computer Fundamentals &amp; Problem-Solving using C++</b>		15
<b>Textbooks:</b>		
<ol style="list-style-type: none"> <li>1. Rajaraman V., "Fundamentals of Computers", Prentice-Hall of India.</li> <li>2. Norton P., "Introduction to Computers", McGraw Hill Education.</li> <li>3. Goel A., "Computer Fundamentals", Pearson.</li> <li>4. James R. Rambaugh, "Object Oriented Design and Modeling", PHI.</li> <li>5. Booch Grady, "Object Oriented Analysis and Design with Application", Pearson, 3rded.</li> <li>6. Dillon and Lee, "Object Oriented Conceptual Modeling", New Delhi PHI-1993.</li> <li>7. Stephen R. Shah, "Introduction to Object Oriented Analysis and Design", TMH.</li> <li>8. Berzin Joseph, "Data Abstraction; The Object-Oriented Approach Using C++", McGraw Hill.</li> <li>9. Herbert Schildt, "C++: The Complete Reference", McGraw Hill, 4thed., 2003.</li> <li>10. Walter Savitch, "Absolute C++", Pearson, 5thed., 2012.</li> <li>11. Lipman, Stanley B, Jonsce Lajole, "C++ Primer Reading", AWL-1999</li> <li>12. Bjarne Stroustrup, "The C++ Programming Language", Pearson, 3rded., 2002.</li> <li>13. E. Balagurusamy, "Object Oriented Programming with C++", TMH, 6thed., 2013.</li> </ol>		

Subject: Computer Science			
Programme/Class: Certificate	Year: 1 <sup>st</sup>	Semester: I	Course Type: GE 1
Course Code:	Course Title: Fundamental of Computers	Credits: L   T   P (3   1   0)	
<b>Course outcomes:</b> <ul style="list-style-type: none"> <li>Understand different components of a computer.</li> <li>Differentiate between hardware and software.</li> <li>Learn the basic concepts of operating systems, networking, internet.</li> <li>Understand various advanced and emerging technologies.</li> </ul>			
<b>Course Prerequisites:</b> Basic Knowledge of Computers			
<b>Course Outline:</b>			
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	
<b>Textbooks:</b> <ol style="list-style-type: none"> <li>Rajaraman V., “Fundamentals of Computers”, Prentice-Hall of India.</li> <li>Norton P., “Introduction to Computers”, McGraw Hill Education.</li> <li>Goel A., “Computer Fundamentals”, Pearson.</li> <li>Balagurusamy E., “Fundamentals of Computers”, McGraw Hill</li> <li>Thareja R., “Fundamentals of Computers”, Oxford University Press.</li> <li>Bindra J., “The Tech Whisperer- on Digital Transformation and the Technologies that Enable it”, Penguin</li> </ol>			

<b>Subject:</b> Computer Science			
<b>Programme/Class:</b> Certificate	<b>Year:</b> 1 <sup>st</sup>	<b>Semester:</b> II	<b>Course Type:</b> DSC 2
<b>Course Code:</b>	<b>Course Title:</b> Data Structures	<b>Credits:</b> L   T   P (3   0   1)	
<b>Course outcomes:</b>			
<ul style="list-style-type: none"> <li>• Understand different problem-solving techniques</li> <li>• Differentiate between sequential lists and linked lists.</li> <li>• Learn and implement searching and sorting techniques.</li> <li>• Understand the fundamentals and application of trees.</li> <li>• Describe the graph terminologies and applications.</li> </ul>			
<b>Course Prerequisites:</b> Proficiency in a programming language like C++ or Java.			
<b>Course Outline:</b>			
<b>Unit</b>	<b>Topics</b>	<b>No. of Lab/Lectures</b>	
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	
<b>Lab: Data Structures</b>			<b>15</b>
<b>Textbooks:</b>			
<ol style="list-style-type: none"> <li>1. Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum, “Data Structures using C and C++”, 2nded., Pearson Education, 2006.</li> <li>2. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, “Fundamentals of Data Structures in C”, Universities press, 2nded., 2008.</li> <li>3. Robert Sedgewick, Kevin Wayne, “Algorithms”, Pearson Education, 4thed., 2011.</li> <li>4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, “Introduction to Algorithms”, PHI, 3rded., 2010.</li> <li>5. Seymour Lipschutz, “Data Structures”, Schaum’s outlines, McGraw Hill Education, 1sted., 2014.</li> <li>6. Donald E. Knuth, “The Art of Computer Programming”, Vol. 1 and Vol. 3.</li> </ol>			

**Subject: Computer Science**

<b>Programme/Class:</b> Certificate		<b>Year:</b> 1 <sup>st</sup>	<b>Semester:</b> II	<b>Course Type:</b> GE 2
<b>Course Code:</b>		<b>Course Title:</b> Cyber Security Awareness	<b>Credits:</b> L   T   P (3   1   0)	
<b>Course outcomes:</b>				
<ul style="list-style-type: none"> <li>Understand the fundamentals of cyber security and its importance.</li> <li>Identify common cyber threats and vulnerabilities.</li> <li>Learn best practices for cyber security and risk management.</li> <li>Understand relevant cyber laws and ethical considerations.</li> <li>Develop practical skills to protect against cyber-attacks.</li> </ul>				
<b>Course Prerequisites:</b> Basic Knowledge of Computers				
<b>Course Outline:</b>				
<b>Unit</b>	<b>Topics</b>			<b>No. of Lab/Lectures</b>
I.	<b>Introduction to Cyber Security:</b> 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.	<b>Cyber Threats and Attacks</b> 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.	<b>Cyber Security Best Practices</b> 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.	<b>Cyber Security Tools and Technologies</b> 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.	<b>Cyber Security Policies and Compliance</b> 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.	<b>Cyber Laws and Ethics</b> 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
	<b>Tutorial:</b> <ul style="list-style-type: none"> <li>Hands-on Cyber Security Practices: Installing and configuring antivirus software: Hands-on experience with installing, configuring, and updating antivirus software to protect against malware.</li> </ul>			15

- |  |   |  |
|--|---|--|
|  | <ul style="list-style-type: none"><li>• Implementing basic firewall rules: Configuring firewall rules to filter network traffic and protect against unauthorized access.</li><li>• Conducting phishing simulations and awareness training: Practical exercises to recognize phishing emails and educate users about phishing threats.</li></ul> |  |
|--|---|--|

**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: 2 <sup>nd</sup>	Semester: III	Course Type: DSC 3
Course Code:	Course Title: Digital System Design	Credits: L   T   P (3   1   0)	
<b>Course outcomes:</b>			
<ul style="list-style-type: none"> <li>Understand the Boolean expressions and their realizations.</li> <li>Design combinational and sequential building blocks.</li> <li>Use these building blocks to design digital circuits.</li> <li>Learn Verilog to design/model digital system.</li> </ul>			
<b>Course Prerequisites:</b> Basic Knowledge of Computers			
<b>Course Outline:</b>			
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	
<b>Textbooks:</b>			
<ol style="list-style-type: none"> <li>M. Morris Mano, Digital Logic and Computer Design, Pearson Education, 1sted., 2004.</li> <li>M. Morris Mano, Michael D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, Pearson Education, 5thed., 2014.</li> </ol>			
<b>References:</b>			
<ol style="list-style-type: none"> <li>David A. Patterson, John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann, 5thed., 2016.</li> <li>M. Morris Mano, Computer System Architecture, Pearson Education, 3rded., 2008.</li> <li>John F. Wakerly, Digital Design Principles and Practices, Pearson Education, 4thed., 2007.</li> <li>Charles H. Roth Jr, Fundamentals of Logic Design, Jaico Publishing House, 5thed., 2003.</li> <li>5. Donald D. Givone, Digital Principles and Design, Tata McGraw Hill, 2003.</li> </ol>			

<b>Subject: Computer Science</b>			
<b>Programme/Class:</b> Diploma	<b>Year:</b> 2 <sup>nd</sup>	<b>Semester:</b> III	<b>Course Type:</b> GE 3/DSE 1
<b>Course Code:</b>	<b>Course Title:</b> Software Engineering	<b>Credits:</b> L   T   P (3   1   0)	
<b>Course outcomes:</b>			
<ul style="list-style-type: none"> <li>• Understand the terminologies of software engineering paradigms</li> <li>• Describe the software engineering requirements and metrics</li> <li>• Analyze the software development life cycle.</li> <li>• Explain the software maintenance and current trends in software engineering</li> <li>• Demonstrate the Computer Aided Software Engineering (CASE) tools</li> </ul>			
<b>Course Prerequisites:</b> Basic programming skills and understanding of software development concepts.			
<b>Course Outline:</b>			
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	
<b>Textbooks:</b>			
<ol style="list-style-type: none"> <li>1. K.K. Aggarwal, Yogesh Singh, "Software Engineering", New Age International, 2<sup>nd</sup>ed., 2005.</li> <li>2. R.S. Pressman, "Software Engineering – A practitioner's approach", McGraw Hill, 5<sup>th</sup>ed., 2001.</li> <li>3. Stephen R. Schach, "Classical &amp; Object Oriented Software Engineering", IRWIN, 1996</li> <li>4. James Peter, W. Pedrycz, "Software Engineering: An Engineering Approach", John Wiley &amp; Sons.</li> <li>5. I. Sommerville, "Software Engineering", Addison Wesley, 2002.</li> </ol>			

<b>Subject: Computer Science</b>			
<b>Programme/Class:</b> Diploma	<b>Year:</b> 2 <sup>nd</sup>	<b>Semester:</b> IV	<b>Course Type:</b> DSC 4
<b>Course Code:</b>	<b>Course Title:</b> Computer System Architecture	<b>Credits:</b> L   T   P (3   1   0)	
<b>Course outcomes:</b>			
<ul style="list-style-type: none"> <li>• To develop understanding of Computer Models and its usage.</li> <li>• To develop understanding of ALU Design.</li> <li>• To conceptualize the understanding of Control Unit design, Memory, IPC, Control Design.</li> <li>• To develop understanding of Memory &amp; Input/output organization Overview.</li> </ul>			
<b>Course Prerequisites:</b> Basic understanding of computer organization.			
<b>Course Outline:</b>			
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	
<b>Textbooks:</b>			
<ol style="list-style-type: none"> <li>1. M. Morris Mano, “Computer System Architecture”, Pearson, 3rded., 2007.</li> <li>2. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, McGrawHill, 5thed., 2002.</li> <li>3. Yu-Cheng Liu, Glenn A. Gibson, “Microcomputer Systems: The 8086 / 8088 Family: Architecture, Programming and Design”, Prentice Hall of India, 2nded., 2007.</li> </ol>			
<b>References:</b>			
<ol style="list-style-type: none"> <li>1. David A. Patterson, John L. Hennessy, “Computer Organization and Design: The Hardware/Software Interface”, Morgan Kaufmann, 3rded., 2007.</li> <li>2. John L. Hennessy, David A. Patterson, “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann, 5thed., 2012.</li> <li>3. John P. Hayes, “Computer Architecture and Organization”, McGraw-Hill Education, 2nded., 1998.</li> <li>4. William Stallings, “Computer Organization and Architecture – Designing for Performance”, Pearson Education, 6thed., 2003.</li> <li>5. Douglas V. Hall, “Microprocessors and Interfacing, Programming and Hardware”, TMH, 2012.</li> </ol>			

<b>Subject: Computer Science</b>			
<b>Programme/Class:</b> Diploma	<b>Year:</b> 2 <sup>nd</sup>	<b>Semester:</b> IV	<b>Course Type:</b> GE 4/DSE 2
<b>Course Code:</b>	<b>Course Title:</b> Programming in JAVA	<b>Credits:</b> L   T   P (3   0   1)	
<b>Course outcomes:</b>			
<ul style="list-style-type: none"> <li>• Use the syntax and semantics of java programming language and basic concepts of OOP.</li> <li>• Develop reusable programs using the concepts of inheritance, polymorphism, interfaces and packages.</li> <li>• Apply the concepts of Multi-threading and Exception handling to develop efficient and error free codes.</li> <li>• Design event driven GUI Applications.</li> </ul>			
<b>Course Prerequisites:</b> Proficiency in basic programming concepts.			
<b>Course Outline:</b>			
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	
<b>Lab: Programming in JAVA</b>			15
<b>Textbooks:</b>			
<ol style="list-style-type: none"> <li>1. Herbert Schildt, "Java, The Complete Reference", TMH, 7thed., 2007.</li> <li>2. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4thed., 2005.</li> <li>3. Cay S. Horstmann, GaryCornell, "Core Java 2 Volume 1 and 2", Prentice Hall, 9thed., 2012.</li> <li>4. Bruce Eckel, "Thinking in Java", PHI, 3rded., 2002.</li> <li>5. Paul Deitel, Harvey Deitel, "Java: How to Program", Prentice Hall, 10thed., 2011.</li> <li>6. Bert Bates and Kathy Sierra "Head First Java", Orielly Media Inc., 2nded., 2005.</li> <li>7. Elliotte Rusty Harold, "Java Network Programming", O'Reilly publishers, 2000</li> <li>8. Ed Roman, "Mastering Enterprise Java Beans", John Wiley &amp; Sons Inc., 1999.</li> </ol>			

<b>Subject: Computer Science</b>			
<b>Programme/Class:</b> Diploma	<b>Year:</b> 2 <sup>nd</sup>	<b>Semester:</b> IV	<b>Course Type:</b> GE 5/DSE 3
<b>Course Code:</b>	<b>Course Title:</b> C# with .NET Framework	<b>Credits:</b> L   T   P (3   0   1)	
<b>Course outcomes:</b>			
<ul style="list-style-type: none"> <li>• Acquire the knowledge of the structure and model of the programming language C #</li> <li>• Understand the use of programming language C # for various programming technologies</li> <li>• Evaluate user requirements for software functionality required to decide whether the programming language C # can meet user requirements</li> <li>• Develop variety of software in C #</li> </ul>			
<b>Course Prerequisites:</b> Proficiency in basic programming concepts.			
<b>Course Outline:</b>			
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	
<b>Lab: C# with .NET Framework</b>			<b>15</b>
<b>Textbooks:</b>			
<ol style="list-style-type: none"> <li>1. Jeffrey Richter, "Applied Microsoft .NET Framework Programming", (Microsoft)</li> <li>2. Fergal Grimes, "Microsoft .Net for Programmers", (SPD)</li> <li>3. Balagurusamy, "Programming with C# ", TMH</li> <li>4. Wiley," Beginning Visual C# 2008",Wrox</li> </ol>			

<b>Subject: Computer Science</b>			
<b>Programme/Class:</b> B.Sc.	<b>Year:</b> 3 <sup>rd</sup>	<b>Semester:</b> V	<b>Course Type:</b> DSC 5
<b>Course Code:</b>	<b>Course Title:</b> Database Management System	<b>Credits:</b> L   T   P (3   0   1)	
<b>Course outcomes:</b>			
<ul style="list-style-type: none"> <li>• Demonstrate a clear understanding of the basics of Database and its use.</li> <li>• Implement Relational Model for Industry as well for all organizations</li> <li>• Understanding Normalization for fast access of records as well transactions</li> </ul>			
<b>Course Prerequisites:</b> Basic understanding of databases and SQL.			
<b>Course Outline:</b>			
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	
<b>Lab: Database Management System</b>			<b>15</b>
<b>Textbooks:</b>			
<ol style="list-style-type: none"> <li>1. Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database Systems", Pearson Education, 5th ed., 2008.</li> <li>2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database Concepts", McGraw-Hill, 6th ed., 2013.</li> <li>3. R. Ramakrishanan, J. Gehrke, "Database Management Systems" McGraw-Hill, 3rd ed., 2002.</li> <li>4. Peter Rob, Carlos Coronel, "Database Systems: Design, Implementation and Management", 7th ed., 2006.</li> </ol>			

**Subject:** Computer Science**Programme/Class:** B.Sc. **Year:** 3<sup>rd</sup> **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.	<b>Introduction to Python:</b> 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.	<b>Data Structures in Python:</b> Lists, tuples, dictionaries, Sets and their operations, Working with files	9
III.	<b>Object-Oriented Programming in Python:</b> Classes and objects, Inheritance and polymorphism, Encapsulation and abstraction	9
IV.	<b>Advanced Python Programming:</b> Exception handling, Decorators and generators, Regular expressions	9
V.	<b>Database Connectivity in Python:</b> Overview of database concepts, Connecting Python with databases (SQLite, MySQL, PostgreSQL), Performing CRUD operations (Create, Read, Update, Delete) using Python	9
<b>Lab: Programming in Python</b>		<b>15</b>

**Textbooks:**

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

Subject: Computer Science			
Programme/Class: B.Sc.	Year: 3 <sup>rd</sup>	Semester: VI	Course Type: DSC 6
Course Code:	Course Title: Operating System	Credits: L   T   P (3   1   0)	
<b>Course outcomes:</b> <ul style="list-style-type: none"> <li>Understand concept of different type of Operating Systems</li> <li>Understand the Program, Processes difference and used corresponding to devices.</li> <li>Understand the efficient memory utilization with file management</li> </ul>			
<b>Course Prerequisites:</b> DSC 1/DSC 3/DSE 4			
<b>Course Outline:</b>			
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	
<b>Textbooks:</b> <ol style="list-style-type: none"> <li>Abraham Silberschatz, Peter B. Galvin, Greg Gagne, “Operating System Concepts”, John Wiley Publications, 8thed., 2008.</li> <li>William Stallings, “Operating Systems: Internals and Design Principles”, Pearson, 7thed., 2013.</li> </ol>			

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: 3 <sup>rd</sup>	Semester: VI	Course Type: GE 7/DSE 5
Course Code:	Course Title: Data Analysis & Visualization using Python	Credits: L   T   P (3   0   1)	
<b>Course outcomes:</b> <ul style="list-style-type: none"> <li>• Apply descriptive statistics to obtain a deterministic view of data</li> <li>• Perform data handling using Numpy arrays</li> <li>• Load, clean, transform, merge and reshape data using Pandas</li> <li>• Visualize data using Pandas and matplotlib libraries</li> </ul>			
<b>Course Prerequisites:</b> Proficiency in Python programming.			
<b>Course Outline:</b>			
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	
<b>Lab: Data Analysis &amp; Visualization using Python</b>			15
<b>Textbooks:</b> <ol style="list-style-type: none"> <li>1. "Python for Data Analysis" by Wes McKinney</li> <li>2. "Interactive Data Visualization with Python" by Bokeh Development Tea</li> </ol>			
<b>Software Requirement:</b> <ol style="list-style-type: none"> <li>1. Python (latest version)</li> <li>2. Jupyter Notebook (for coding and documentation)</li> <li>3. Required libraries (pandas, NumPy, matplotlib, Seaborn, Plotly)</li> </ol>			