INFROMATION SCIENCE AND ENGINEERING

SCHEME and SYLLABUS

OF

VII and VIII SEMESTER B.E.

FOR THE A.Y: 2025-26

SCHEME OF TEACHING AND EXAMINATION (2022 Scheme) (w.e.f. 2025-26)

VII Semester (Swappable VII and VIII Semester)

Sl. No.	Course and Course Code		Course Title	Teaching / Paper setting Dept.	Teaching hrs./week				Examination				Credits
					Lecture	Tutorial	Practical/ Drawing	Self-Study Component	Duration in hrs.	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1.	IPCC	S7ISI01	Fullstack Development (Integrated)		3	0	2		3	50	50	100	4
2.	IPCC	S7ISI02	Software Testing (Integrated)		3	0	2		3	50	50	100	4
3.	PCC	S7IS01	Cyber Security and Digital Forensics		3	2	0		3	50	50	100	4
4.	PEC	S7ISPEC3X	Professional Elective Course-III		3	0	0		3	50	50	100	3
5.	OEC		Open Elective Course-II		3	0	0		3	50	50	100	3
6.	PROJ	S7ISMP2	Major Project Phase II		0	0	12		3	100	100	200	6
			Total							**350**	**350**	**700**	**24**
		AAP	AICTE Activity Points (Applicable for both Regular and Lateral Entry students)	40 hours community service to be documented and produced for the examination									

Note: **IPCC**: Integrated Professional Core Course, **PCC**: Professional Core Course; **PEC**: Professional Elective Course**;**
OEC: Open Elective Course; **PROJ:** Project Phase –II;
L: Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation.

Professional Elective Course (PEC) (Offered by the Department)

S7ISPEC031	Blockchain Technology	S7ISPEC032	Advanced Data Structures
S7ISPEC033	Software Defined Networks	S7ISPEC034	Agile Software Technology and DevOps
S7ISPEC035	Quantum COmputing		

Note: VII and VIII semesters of IV years of the program

1) Institutions can swap the VII and VIII Semester Schemes of Teaching and Examinations to accommodate research internships/ industry internships after the VI semester.
2) Credits earned for the courses of VII and VIII Semester Scheme of Teaching and Examinations shall be counted against the corresponding semesters whether the VII or VIII semesters is completed during the beginning of the IV year or the later part of IV years of the program.

Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical of the same course. Credit for IPCC can be 04 and its Teaching–Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor

of Engineering (B.E.) 2022-23 may please be referred.

Professional Elective Courses (PEC): A professional elective (PEC) course is intended to enhance the depth and breadth of educational experience in the Engineering and Technology curriculum. Multidisciplinary courses that are added supplement the latest trend and advanced technology in the selected stream of Engineering. Each group will provide an option to select one course. The minimum number of students' strengths for offering a professional elective is 10. However, this conditional shall not be applicable to cases where the admission to the program is less than 10.

Open Elective Courses:

Students belonging to a particular stream of Engineering and Technology are not entitled to the open electives offered by their parent Department. However, they can opt for an elective offered by other Departments, provided they satisfy the prerequisite condition if any. Registration to open electives shall be documented under the guidance of the Program Coordinator/ Advisor/Mentor. The minimum numbers of students' strength for offering Open Elective Course is 10. However, this condition shall not be applicable to class where the admission to the program is less than 10.

Project Work: The objective of the Project work is

i) To encourage independent learning and the innovative attitude of the students.

ii) To develop interactive attitude, communication skills, organization, time management, and presentation skills.

iii) To impart flexibility and adaptability.

iv) To inspire team working.

v) To expand intellectual capacity, credibility, judgment and intuition.

vi) To adhere to punctuality, setting and meeting deadlines.

vii) To install responsibilities to oneself and others.

viii) To train students to present the topic of project work in a seminar without any fear, face the audience confidently, enhance communication skills, involve in group discussion to present and exchange ideas.

CIE procedure for Project Work:

1) Single discipline: The CIE marks shall be awarded by a committee consisting of the Head of the concerned Department and two senior faculty members of the Department, one of whom shall be the Guide.

   The CIE marks awarded for the project work, shall be based on the evaluation of the project work Report, project presentation skill, and question and answer session in the ratio 50:25:25. The marks awarded for the project report shall be the same for all the batch mates.

2) Interdisciplinary: Continuous Internal Evaluation shall be group-wise at the college level with the participation of all guides of the college. Participation of external guide/s, if any, is desirable. The CIE marks awarded for the project work, shall be based on the evaluation of project work Report, project presentation skill, and question and answer session in the ratio 50:25:25. The marks awarded for the project report shall be the same for all the batch mates.

SEE procedure for Project Work: SEE for project work will be conducted by the two examiners appointed by the University. The SEE marks awarded for the project work shall

be based on the evaluation of project work Report, project presentation skill, and question and answer session in the ratio 50:25:25.

SCHEME OF TEACHING AND EXAMINATION (2022 Scheme) (w.e.f. 2025-26)

VIII Semester (Swappable VII and VIII Semester)

Sl. No.	Course and Course Code		Course Title	Teaching / Paper setting Dept.	Teaching hrs./week				Examination				Credits
					Lecture	Tutorial	Practical/ Drawing	Self-Study Component	Duration in hrs.	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1.	PEC		Professional Elective (Online Courses)		3	0	0		3	50	50	100	3
2.	OEC		Open Elective (Online Courses)		3	0	0		3	50	50	100	3
3.	INT		Internship (Industry/Research) (14-20 weeks)		0	0	12		3	100	100	200	10
			Total							**200**	**200**	**400**	**16**
		AAP	AICTE Activity Points (Applicable for both Regular and Lateral Entry students)	40 hours community service to be documented and produced for the examination									

Note: **PEC**: Professional Elective Course**; OEC:** Open Elective Course (Online); **INT:** Industry Internship / Research Internship / Rural Internship
L: Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation.

Professional Elective (Online Courses – suggested by BoS, NPTEL)			
	Computational Complexity		Social Networks
	Privacy and Security in Online Social Media		Foundations of Virtual Reality
	Real-Time Systems		Computer Architecture
Open Elective (Online Courses – suggested by BoS, NPTEL)			
	Introduction to virtual reality		Remote Sensing: Principles and Applications
	Rapid Manufacturing		BioInformatics: Algorithms and Applications
	Robotics		

Note: VII and VIII semesters of IV years of the program

1) Institutions can swap the VII and VIII Semester Schemes of Teaching and Examinations to accommodate research internships/ industry internships after the VI semester.
2) Credits earned for the courses of VII and VIII Semester Scheme of Teaching and Examinations shall be counted against the corresponding semesters whether the VII or VIII semesters is completed during the beginning of the IV year or the later part of IV years of the program.

Elucidation:

At the beginning of IV years of the program i.e., after VI semester, VII semester classwork and VIII semester Research Internship /Industrial Internship / Rural Internship shall be permitted to be operated simultaneously by the University so that students have ample opportunity for an internship. In other words, a good percentage of the class shall attend VII semester classwork and a similar percentage of others shall attend to Research Internship or Industrial Internship or Rural Internship.

Research/Industrial /Rural Internship shall be carried out at an Industry, NGO, MSME, Innovation center, Incubation center, Start-up, center of Excellence (CoE), Study Centre established in the parent institute and /or at reputed research organizations/institutes.

The mandatory Research internship /Industry internship / Rural Internship is for 14 to 20 Weeks. The internship shall be considered as a head of passing and shall be considered for the award of a Degree. Those, who do not take up/complete the internship shall be declared to fail and shall have to complete it during the subsequent University examination after satisfying the internship requirements.

Research internship: A research internship is intended to offer the flavor of current research going on in the research field. It helps students get familiarized with the field and imparts the skill required for carrying out research.

Industry internship: Is an extended period of work experience undertaken by students to supplement their Degree for professional development. It also helps them learn to overcome unexpected obstacles and successfully navigate organizations, perspectives, and cultures. Dealing with contingencies helps students recognize, appreciate, and adapt to organizational realities by tempering their knowledge with practical constraints.

Rural Internship: Rural development internship is an initiative of Unnat Bharat Abhiyan Cell, RGIT in association with AICTE to involve students of all departments studying in different academic years for exploring various opportunities in techno-social fields, to connect and work with Rural India for their upliftment.

The faculty coordinator or mentor has to monitor the student's internship progress and interact with them to guide for the successful completion of the internship.

The students are permitted to carry out the internship anywhere in India or abroad. University shall not bear any expenses incurred in respect of the internship.

With the consent of the internal guide and Principal of the Institution, students shall be allowed to carry out the internship at their hometown (within or outside the state or abroad), provided favorable facilities are available for the internship and the student remains regularly in contact with the internal guide. University shall not bear any cost involved in carrying out the internship by students. However, students can receive any financial assistance extended by the organization.

Professional Elective /Open Elective Course: These are ONLINE courses suggested by the respective Board of Studies. Details of these courses shall be made available for students on the VTU web portal.

FULL STACK DEVELOPMENT

Contact Hours/ Week:	3L+2P	Credits:	4
Total Lecture Hours:	42	CIE Marks:	50
Total Practical Hours:	26	SEE Marks:	50
Course Code:	S7ISI01		

Course objectives: This course will enable students to:	
1	Understand the essential javascript concepts for web development.
2	Style Web applications using bootstrap.
3	Utilize React JS to build front end User Interface.
4	Understand the usage of API's to create web applications using Express JS.
5	Store and model data in a noSQL database.

UNIT I
Basic JavaScript Instructions, Statements, Comments, Variables, Data Types, Arrays, Strings, Functions, Methods & Objects, Decisions & Loops. TB 1: Chapter 2, 3, 4
8 Hours

UNIT II
Document Object Model: DOM Manipulation, Selecting Elements, Working with DOM Nodes, Updating Element Content & Attributes, Events, Different Types of Events, How to Bind an Event to an Element, Event Delegation, Event Listeners. TB 1: Chapter: 5, 6, 13
8 Hours

UNIT III
Form enhancement and validation. Introduction to MERN: MERN components, Server less Hello world. React Components: Issue Tracker, React Classes, Composing Components, Passing Data Using Properties, Passing Data Using Children, Dynamic Composition. TB 2: Chapter 1, 2, 3
8 Hours

UNIT IV
React State: Initial State, Async State Initialization, Updating State, Lifting State Up, Event Handling, Stateless Components, Designing Components, State vs. Props, Component Hierarchy, Communication, Stateless Components. Express, REST API, GraphQL, Field Specification, Graph Based, Single Endpoint, Strongly Typed, Introspection, Libraries, The About API GraphQL Schema File, The List API, List API Integration, Custom Scalar types, The Create API, Create API Integration, Query Variables, Input Validations, Displaying Errors.

TB 2: Chapter 4, 5
9 Hours

UNIT V
MongoDB: Basics, Documents, Collections, Databases, Query Language, Installation, The Mongo Shell, MongoDB CRUD Operations, Create, Read, Projection, Update, Delete, Aggregate, MongoDB Node.js Driver, Schema Initialization, Reading from MongoDB, Writing to MongoDB. Modularization and Webpack, Back-End Modules Front-End Modules and Webpack Transform and Bundle, Libraries Bundle, Hot Module Replacement, Debugging DefinePlugin: Build Configuration, Production Optimization. TB 2: Chapter 6, 7
9 Hours

Laboratory Experiments	
1.	a. Write a script that Logs "Hello, World!" to the console. Create a script that calculates the sum of two numbers and displays the result in an alert box. b. Create an array of 5 cities and perform the following operations: Log the total number of cities. Add a new city at the end. Remove the first city. Find and log the index of a specific city.
2.	Read a string from the user, Find its length. Extract the word “JavaScript” using substring() or slice(). Replace one word with another word and log the new string. Write a function isPalindrome(str) that checks if a given string is a palindrome (reads the same backward).
3.	Create an object student with properties: name (string), grade (number), subjects (array), displayInfo() (method to log the student's details). Write a script to dynamically add a passed property to the student object, with a value of true or false based on their grade. Create a loop to log all keys and values of the student object.
4.	Create a button in your HTML with the text "Click Me". Add an event listener to log “Button clicked!” to the console when the button is clicked. Select an image and add a mouseover event listener to change its border color. Add an event listener to the document that logs the key pressed by the user.
5.	Build a React application to track issues. Display a list of issues (use static data). Each issue should have a title, description, and status (e.g., Open/Closed). Render the list using a functional component.
6.	Create a component Counter with A state variable count initialized to 0. Create Buttons to increment and decrement the count. Simulate fetching initial data for the Counter component using useEffect (functional component) or componentDidMount (class component). Extend the Counter component to Double the count value when a button is clicked. Reset the count to 0 using another button.
7.	Install Express (npm install express). Set up a basic server that responds with “Hello, Express!” at the root endpoint (GET /). Create a REST API. Implement endpoints for a Product resource: GET: Returns a list of products. POST: Adds a new product. GET /:id: Returns details of a specific product. PUT /:id: Updates an existing product. DELETE /:id: Deletes a product. Add middleware to log requests to the console. Use express.json() to parse incoming JSON payloads.
8.	Install the MongoDB driver for Node.js. Create a Node.js script to connect to the shop database. Implement insert, find, update, and delete operations using the Node.js MongoDB driver.

	Define a product schema using Mongoose. Insert data into the products collection using Mongoose. Create an Express API with a /products endpoint to fetch all products. Use fetch in React to call the /products endpoint and display the list of products. Add a POST /products endpoint in Express to insert a new product. Update the Product List, after adding a product, update the list of products displayed in React.
	28 Hours

TEXT BOOKS		
1	Jon Duckett	JavaScript & jQuery: Interactive Front-End Web Development, Wiley Publication, 2014
2	Vasan Subramanian	Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node. Apress Publication, 2019.

REFERENCES		
1	Vasan Subramanian	https://github.com/vasansr/pro-mern-stack
2	Prof. Gaurav Raina, Tanmai Gopal	Introduction to Modern Application Development, IIT Madras
3	IIT, Kharagpur	https://archive.nptel.ac.in/courses/106/105/106105084/

Course Outcomes: Upon completion of this course the student will be able to:	
CO1	**Apply** Javascript to build dynamic and interactive Web projects.
CO2	**Implement** user interface components for JavaScript-based Web using React.JS.
CO3	**Apply** Express Node to build web applications on the server side.
CO4	**Develop** data models in an open source NoSQL database.
CO5	**Demonstrate** modularization and packing of the front-end modules.

Course Articulation Matrix

		POs											**PSOs**		
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
COs	**CO1**	3											2		
	CO2			2									2		
	CO3	3											2		
	CO4			2									2		
	CO5			3		3							2		2

SOFTWARE TESTING (Integrated)

Contact Hours/ Week:	3L+2P	Credits:	4
Total Lecture Hours:	42	CIE Marks:	50
Total Practical Hours:	26	SEE Marks:	50
Course Code:			

Course objectives: This course will enable students to:	
1.	Understand the fundamental concepts of software testing.
2.	Explain the life cycle of the testing and explore various levels of testing.
3.	Gain knowledge on functional and structural testing methods.
4.	Explore the concepts of software testing and its application in various scenarios with the help different testing strategies.
5.	Understand and use testing tools and frameworks for various testing tasks.

UNIT I
A Perspective on Testing, Examples: Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Levels of testing, Examples: Generalized pseudo code, The triangle problem, The NextDate function, The Foodies-Wish-List Online Shopping Application, The Garage Door Controller, The currency converter, Saturn windshield wiper. **Boundary Value Testing:** Boundary value analysis, Robustness testing, Worst-case testing, Special value testing, Examples, Random Testing, Guidelines for Boundary value testing.
9 Hours

UNIT II
The Equivalence Class Testing, Decision Table-Based Testing: Equivalence classes, Traditional equivalence class testing, Improved Equivalence Class Testing, Equivalence test cases for the triangle problem, NextDate function, Equivalence Class Test Cases for the complete Order Method, Edge Testing, Guidelines and observations. **Decision tables,** Decision Table techniques, Test cases for the triangle problem NextDate function, Cause and Effect Graph, Guidelines and observations.
9 Hours

UNIT III
Code based testing **Program Graphs,** DD paths, Code coverage metrics, E. F. Miller's Coverage Metrics, Basis path testing, guidelines and observations. Testing Object-Oriented Software: Unit Testing Frameworks, Mock Objects and Automated Object Mocking, Data Flow Testing, Definition-Use testing, Define/Use Testing, Examples, Object-Oriented Complexity Metrics, Issues in Testing Object-Oriented Software, Slice-Based Testing,
8 Hours

UNIT IV
Beyond Unit Testing: Life Cycle-Based Testing, Traditional Waterfall Testing, Testing in Iterative Lifecycles, Agile Testing, **Software Testing Integration Level:** Software Testing Fundamentals, integration Testing, Artificial Intelligence and Regression Testing, integration Testing in the OO Context, Validation Testing, Testing patters. (Text Book 2: Chapter 20)
8 Hours

UNIT V
Automation Testing Tools: 1. Selenium–Open-source tool for automating web browsers, 2. Katalon Studio Automation tool supporting web, API, mobile, and desktop apps. 3. Appium – Open-source tool for mobile application testing. 4. Apache JMeter – Open-source tool for performance and load testing. 5. JUnit – Popular unit testing framework for Java. 6. TestSigma - NLP-based test creation, self-healing tests, impact analysis
8 Hours

TEXT BOOKS		
1	Paul C. Jorgensen and Byron DeVries	Software Testing, A Craftsman's Approach, , CRC Press, 5th Edition, 2021
2	Roger. S. Pressman, Bruce R. Maxim	Software Engineering - A Practitioners approach, Tata-McGraw Hill. 9th Edition, 2020, (Unit V)
3	Web Links for Tools:	1. https://www.selenium.dev/documentation/ 2. https://katalon.com/katalon-studio 3. https://appium.io/docs/en/latest/ 4. https://jmeter.apache.org/usermanual/ 5. https://junit.org/junit5/docs/current/user-guide/ 6. https://testsigma.com/docs/

REFERENCE BOOKS		
1	Boris Beizer	Software Testing techniques, Dreamtech publication 2018
2	Aditya P Mathur,	Foundations of Software Testing, Pearson Education, 2008.
3	NPTEL Link	https://onlinecourses.nptel.ac.in/noc18_cs42/

Lab exercises:

	Part A **Manual Software Testing**
1	Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene

	triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on **boundary-value analysis** and execute the test cases and discuss the results. Test Data: Enter the 3 Integer Value (a, b and c) Pre-condition: $1 \leq a \leq 10$, $1 \leq b \leq 10$ and $1 \leq c \leq 10$ and $a < b + c$, $b < a + c$ and $c < a + b$
2	Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on and **equivalence class** partitioning, execute the test cases and discuss the results. Test Data: Enter the 3 Integer Value (a , b and c) Pre-condition: $1 \leq a \leq 10$, $1 \leq b \leq 10$ and $1 \leq c \leq 10$ and $a < b + c$, $b < a + c$ and $c < a + b$
3	Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on **decision tables-based** testing, execute the test cases and discuss the results. Test Data: Enter the 3 Integer Value (a , b and c) Pre-condition: $1 \leq a \leq 10$, $1 \leq b \leq 10$ and $1 \leq c \leq 10$ and $a < b + c$, $b < a + c$ and $c < a + b$
4	Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of **boundary-value analysis testing**, derive different test cases, execute these test cases and discuss the test results. Test data : Enter the date with month/ day/ year format Pre-condition: Month 1 to 12, Day 1 to 31 and Year 1812 to 2014
5	Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of **Equivalence partition testing**, derive different test cases, execute these test cases and discuss the test results. Test data: Enter the date with month/ day/ year format Pre-condition: Month 1 to 12, Day 1 to 31 and Year 1812 to 2014
6	Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of **decision table-based testing,** derive different test cases, execute these test cases and discuss the test results. Test data: Enter the date with month/ day/ year format Pre-condition: Month 1 to 12, Day 1 to 31 and Year 1812 to 2014
7	Design, develop, code and run the program in suitable language: Foodies-Wish-List is an online shopping service for extremely rare gourmet foods. It can be used either on a one-time basis as a guest, or repeatedly by members. There is no initial cost for either category, but to be a Foodies-Wish-List member, one must register with customary information, such as: • member name, address, shipping address, telephone number, email address, preferred payment method, member credit card and PayPal Registered Foodies-Wish-List members receive discounts based on the price of an individual order as follows: a) orders less than $200 receive no discount b) orders between $200 and $800 (inclusive) receive a 10% discount c) orders over $800 receive a 15% discount There is no discount for any guest order.

	Foodies-Wish-List members receive free shipping on any order over $200. For orders less than $200, there is a standard shipping price of $5.00. Analyze it from the perspective of **boundary-value analysis testing**, derive different test cases, execute these test cases and discuss the test results.
8	Design, develop, code and run the program in suitable language: Foodies-Wish-List is an online shopping service for extremely rare gourmet foods. It can be used either on a one-time basis as a guest, or repeatedly by members. There is no initial cost for either category, but to be a Foodies-Wish-List member, one must register with customary information, such as: • member name, address, shipping address, telephone number, email address, preferred payment method, member credit card and PayPal Registered Foodies-Wish-List members receive discounts based on the price of an individual order as follows: d) orders less than $200 receive no discount e) orders between $200 and $800 (inclusive) receive a 10% discount f) orders over $800 receive a 15% discount There is no discount for any guest order. Foodies-Wish-List members receive free shipping on any order over $200. For orders less than $200, there is a standard shipping price of $5.00. Analyze it from the perspective of **Equivalence partition testing**, derive different test cases, execute these test cases and discuss the test results.
9	Design, develop, code and run the program in suitable language: Foodies-Wish-List is an online shopping service for extremely rare gourmet foods. It can be used either on a one-time basis as a guest, or repeatedly by members. There is no initial cost for either category, but to be a Foodies-Wish-List member, one must register with customary information, such as: • member name, address, shipping address, telephone number, email address, preferred payment method, member credit card and PayPal Registered Foodies-Wish-List members receive discounts based on the price of an individual order as follows: g) orders less than $200 receive no discount h) orders between $200 and $800 (inclusive) receive a 10% discount i) orders over $800 receive a 15% discount There is no discount for any guest order. Foodies-Wish-List members receive free shipping on any order over $200. For orders less than $200, there is a standard shipping price of $5.00. Analyze it from the perspective of **decision table-based testing**, derive different test cases, execute these test cases and discuss the test results.
	Automation Testing
1	Automate basic browser interactions and validate webpage behavior.---Selenium
2	Automate web and mobile testing using a GUI-based tool.-- Katalon Studio – Cross-platform Automation
3	Automate a basic user interaction (login form) in a native Android application using Appium.
4	Simulate load on web servers and analyze performance using Apache JMeter – Performance Testing tool.
5	Write and run unit tests for Java applications using JUnit – Java Unit Testing Framework

Course Outcomes: Upon completion of this course the student will be able to:	
CO1	**Illustrate** the fundamental concepts, objectives, and importance of software testing in the software development lifecycle.
CO2	**Distinguish** between different levels and types of software testing and apply appropriate techniques for various testing scenarios.
CO3	**Design, develop, and execute** effective test cases and test plans based on software requirements.
CO4	**Apply** different testing methodologies used in industries for software testing.
CO5	**Analyze** and use manual and automated testing tools to efficiently perform functional and non-functional testing.

Course Articulation Matrix

		POs											**PSOs**		
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
COs	**CO1**	2		3									2		
	CO2		2										3		
	CO3			3									2		
	CO4			2									3		
	CO5		2			3				2	2		2		
AVG		2	2	3	2	3				2	2		3		

1: Low association 2: Moderate association 3: High association

Cyber Security and Digital Forensics

Contact Hours/ Week:	3+ 2	Credits:	4.0
Total Lecture Hours:	42	CIE Marks:	50
Course Code:	S7IS01	SEE Marks:	50

Course objectives: This course will enable students to:	
1.	Understand the fundamentals of information systems, the evolution of cybersecurity, and foundational security principles and models.
2.	Explore various types of cybercrimes, their motivations, methods, and global countermeasures
3.	Identify common vulnerabilities in IT systems and assess risk factors impacting data, networks, and third-party dependencies.
4.	Understand the cybersecurity management practices, policies, standards, and business continuity strategies.
5.	Evaluate the role of technologies, legal frameworks, human factors, and emerging trends in building a secure cyber ecosystem.

UNIT I
Security in an Interconnected World: Cyberspace and Cybersecurity, Information Security, Organisation and Governance of the Internet and Cybersecurity, Information Security Models. **Cybercrime, Cyberattack Tools and Methods, Threat Sources and Cyber Offenders:** What is a Cybercrime? Classification of Cybercrimes, Forms of Cybercrimes, Cyber Scams and Frauds, Sources of Threats: Threat Actors and their Motivations, Tools and Methods Used in Cyberattacks/Cybercrimes, what is a Cyberattack? Responding to Cyberattacks and the Cyber Kill Chain, cyberattacks: Organizational Implications, Cyberattacks Impacting Citizens and Communities, Prevention of Cybercrime, International Efforts to Deal with Cybercrime, National Cybersecurity Policy, Online Code of Conduct and Computer Ethics.
8 Hours

UNIT II
Cybersecurity Vulnerabilities: Security Considerations and Challenges, Types of Vulnerabilities, Project OWASP, Vulnerabilities Assessment, Common Vulnerabilities and Exposures (CVE): Institutional Mechanism **Cybersecurity Management Practices**: Overview of Cybersecurity Management, Information Classification Process, Security Policies, Security Procedures and Guidelines, Security Controls, Security Organization, Incident Response, Business Continuity and Disaster Recovery.
8 Hours

UNIT III
Developing Secure Information Systems: Securing Information Assets, Data Security and Protection, Application Security, Security Architecture and Design Security Issues in Hardware, Mobile Devices and Internet of Things, Network Security, Operating system security, Database security, User Management, Physical Security of IT Assets, Techniques/Methods for Data Security and Protection, Issues Related to Digital File Sharing. **Cybersecurity Strategies and Approaches:** Information Security governance and Risk Management, Cyber Risk Management, Cybersecurity Frameworks, Cyber Resilience,

Industry-specific Cybersecurity Frameworks, The Human Factor in Cybersecurity, Algorithms and Techniques for Cybersecurity.
9 Hours

UNIT IV
Cybersecurity Technologies Securing Networks, Web Applications, Services and Servers, Email Security, Antivirus Technologies and Solutions, Identity and Access Management, Authentication, Cryptography How Do Digital Money, Cryptocurrency and NFTs Work?, Digital Signatures, Advanced Technologies and Approaches in Cybersecurity, Internet Protocols and Ports. **Cyber Laws and Forensics** Need for Cyber Laws and Regulations, Role of International Law and Governments, Challenges for Law-makers and Law Enforcement Agencies, Cybersecurity Regulations, Cyber Forensics Cybercrime Techniques, Prevention of Cybercrime and Protection, Cybercrime Investigation, Evidence Collection and Analysis, Intellectual Property Issues in Cyberspace.
9 Hours

UNIT V
Personal Cybersecurity, Privacy and Data Protection: What is Personal Cybersecurity? Common Causes of Personal Security Breaches, Personal Cybersecurity Best Practices, Privacy Regulations and Cybersecurity, The Role of Ethics in Cybersecurity. **Cybersecurity in Evolving Technology and Practice:** Future Challenges in Cybersecurity, Web 3.0, Harnessing Artificial Intelligence for Cybersecurity, Blockchain for Cybersecurity, Quantum Computing and Cybersecurity, Combating Advanced Persistent Threats, Digital Trust and Identity Management, 5G Networks and Cybersecurity, Adopting a 'Secure-by-Design' Approach, Supply Chain Cybersecurity, Other Evolving Aspects of Cybersecurity.
8 Hours

TEXT BOOKS		
1	Ajay Singh	"Introduction to Cybersecurity: Concepts, Principles, Technologies and Practices", University Press, First Edition, 2023.

REFERENCE BOOKS		
1	Henrique M. D. Santos,	Cybersecurity - A Practical Engineering Approach, CRC Press, First edition, 2022.
2	William Stallings, Lawrie Brown	Computer Security Principles and Practice, Pearson ed., 5th edition, 2023.
3	Anas Zakir	Cyber security and Digital Forensics, Clever Fox Publishing, 2022

Course Outcomes: Upon completion of this course the student will be able to:	
CO1	**Describe** the need for information security and the evolution of models and principles governing information systems.
CO2	**Analyze** different forms of cybercrimes, ethical considerations, and apply strategies for cybercrime prevention and digital evidence handling.
CO3	**Identify** and **analyze** vulnerabilities in information systems such as applications, networks, cloud, and third-party ecosystems.

CO4	**Design** cybersecurity strategies using best practices in risk management, policymaking, and incident response aligned with industry frameworks
CO5	**Apply** legal and ethical principles to real-world cybersecurity issues and assess emerging technologies and trends to strengthen cyber resilience.

Course Articulation Matrix

		POs											PSOs		
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
COs	**CO1**	**3**												**2**	
	CO2	**3**	**3**											**3**	
	CO3	**3**	**3**											**2**	
	CO4			**2**										**3**	
	CO5	**3**												**2**	
	Average	**3**	**3**	**2**										**3**	

Tutorials:

1. Use Wireshark to analyze network traffic, look for signs of malicious utilization.
2. Explore Nmap Tool to discover and characterize (scanning) machines in a network.
3. Explore Nessus/ OpenVAS to discover system vulnerabilities.
4. Use of the Metasploit tool, to exploit the identified vulnerabilities.
5. Employ cryptography techniques and tools to protect the information when it is in transit.
6. Explore artifacts that are useful while conducting a forensics investigation. Find where the artifacts are placed and extract valuable information out of them.
7. Investigate techniques for detecting common web application attacks by analyzing logs from web applications and firewalls, identify the attack point, and trace the root cause by pinpointing the exploited vulnerability
8. Experiment on Cracking Passwords
9. Performing SQL Injection on a Test Web Application
10. To simulate a simple DoS attack and observe its impact on a target system or service

Blockchain Technology

Contact Hours/Week	:	3	Credits	:	3.0
Total Lecture Hours	:	42	CIE Marks	:	50
Total Tutorial Hours	:	00	SEE Marks	:	50
Course Code	:	S7ISPEC031			

Course objectives:
This course will enable students to:

1.	Elaborate the fundamental concepts of blockchain.
2.	Comprehend the importance of Game Theory and Cryptography in blockchain.
3.	Interpret the working of different Cryptocurrencies.
4.	Explore various business applications of blockchain.

UNIT I

Introduction to Blockchain: Introduction to Blockchain, Backstory of Blockchain, what is Blockchain? Centralized versus decentralized systems, Centralized Systems, Decentralized Systems, Layers of Blockchain: Application Layer, Execution Layer, Semantic Layer, Propagation Layer, Consensus Layer,
Why is Blockchain Important? Limitations of Centralized Systems, Blockchain Adoption So Far, Blockchain Uses and Use Cases.
T1 Chapter 1

9 Hours

UNIT II

How Blockchain Works: Laying the Blockchain Foundation, Cryptography, Symmetric Key Cryptography, Cryptographic Hash Functions, MAC and HMAC, Asymmetric Key Cryptography, Diffie-Hellman Key Exchange, Symmetric vs. Asymmetric Key Cryptography, Game Theory, Nash Equilibrium, Prisoner's Dilemma, Byzantine Generals' Problem, Zero-Sum Games, Why to Study Game Theory, Computer Science Engineering , The Blockchain , Merkle Trees, Putting It All Together, Properties of Blockchain Solutions.

Blockchain Transactions, Distributed Consensus Mechanisms, Blockchain, Applications, Scaling Blockchain Off-Chain Computation Sharding Blockchain State.

T1 Chapter 2

8 Hours

UNIT III

How Bitcoin works: The History of Money, Dawn of Bitcoin , What Is Bitcoin? Working with Bitcoins , The Bitcoin Blockchain, Block Structure, The Genesis The Bitcoin Network, Network Discovery for a New Node, Bitcoin Transactions, Consensus and Block Mining, Block Propagation, Putting It all Together, Bitcoin Scripts, Bitcoin Transactions Revisited, Scripts, Full Nodes vs. SPVs, Full Nodes, SPVs.
T1 Chapter 3

8 Hours

UNIT IV

How Ethereum Works: From Bitcoin to Ethereum, Ethereum as a Next-Gen Blockchain, Design Philosophy of Ethereum, Enter the Ethereum Blockchain, Ethereum Blockchain, Ethereum Accounts, Trie Usage, Merkle Patricia Tree, RLP Encoding, Ethereum Transaction and Message Structure, Ethereum State Transaction Function, Gas and Transaction Cost,

Ethereum Smart Contracts, Contract Creation, Ethereum Virtual Machine and Code Execution, Ethereum Ecosystem, Swar, Whisper, DApp, Development Components. T1 Chapter 4
8 Hours

UNIT V
Propelling Business with Blockchains, Recognizing Types of Market Friction, Information frictions, Interaction frictions, Innovation frictions, Moving Closer to Friction-Free Business Networks, Reducing information friction, Easing interaction friction, Easing innovation friction, Transforming Ecosystems through Increased Visibility, Blockchain in Action: Use Cases Financial Services, Trade finance, Post-trade clearing and settlement, Cross-border transactions, Trusted digital identity, Multinational Policy Management, Government, Supply Chain Management, Food safety, Global trade, Healthcare, Electronic medical records, Healthcare payment preauthorization T2 - Chapters 3 & 4
9 Hours

TEXT BOOKS		
1	Bikramaditya Singhal, Gautam Dhameja, Privansu Sekhar Panda	Beginning Blockchain, Apress Media, 2018.
2	Manav Gupta	Blockchain For Dummies, John Wiley & Sons, 2nd IBM Limited Edition, 2019.

REFERENCE BOOKS		
1	Peter Lypovonyav	Blockchain for Business 2019, Packt Publishing Limited, 2019
2.	Debajani Mohanty	Ethereum for Architects and Developers, Apress Media, 2018

Course Outcomes: Upon completion of this course the student will be able to:	
CO1:	**Describe** the fundamentals and underlying concepts of working of a Blockchain.
CO2:	**Apply** Cryptography and game theory concepts for blockchain solutions.
CO3:	**Illustrate** the working of bitcoin cryptocurrency.
CO4:	**Outline** the use of Ethereum in building blockchain applications
CO5:	**Describe** the potential use cases of blockchain and **identify** the associated benefits.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes (PSOs).

		POs												PSOs		
		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
COs	CO1	3													3	
	CO2		3												3	
	CO3		2												2	
	CO4	3													3	
	CO5	3													2	
AVG		3	3												3	

ADVANCED DATA STRUCTURES AND ALGORITHMS

Contact Hours/Week	:	3	Credits	:	3.0
Total Lecture Hours	:	42	CIE Marks	:	50
Total Tutorial Hours	:	00	SEE Marks	:	50
Course Code	:	S7ISPEC032			

Course objectives:
This course will enable students to:

1.	Understand the concept of abstract data types and their implementation.
2.	Design different heap structures and perform insert, delete and merge operations
3.	Describe the working of randomized and backtracking algorithm design techniques.
4.	Analyze the performance of different heap structures using amortized analysis.
5.	Develop algorithms to construct advanced data structures like Red-Black trees, AA-trees, and treaps

UNIT I

Abstract Data Types (ADTs): Vector and List in the STL: Iterators, Example: Using **erase** on a list, *const_iterators*, Implementation of vector, implementation of list

Trees : Preliminaries: Implementation of Trees, Tree Traversals with an Application, Binary Trees: Implementation, An Example: Expression Trees, The Search Tree ADT-Binary Search Trees, Splay Trees: A Simple Idea, Splaying, Tree Traversals (Revisited), Sets and Maps in the Standard Library: Sets, Maps, Implementation of *set* and *map*, An Example that uses Several Maps.

9 Hours

UNIT II

Priority Queues (Heaps)
Model, Simple Implementations, Binary Heap: Structural Property, Heap-order Property, Basic Heap Operations, Other Heap Operations, Applications of Priority Queues: The Selection Problem, Event Simulation, d-Heaps, Leftist Heaps: Leftist Heap Property, Leftist Heap Operations, Skew Heaps

8 Hours

UNIT III

Binomial Queues: Binomial Queue Structure, Binomial Queue Operations, Implementation of Binomial Queues, Priority Queues in the Standard Library.

Sorting
Preliminaries, Shell sort, A General Lower Bound for Sorting: Decision Trees, Bucket Sort, External Sorting.

Algorithm Design Techniques
Randomized Algorithms: Random Number Generators, Skip Lists, Primality Testing, Backtracking Algorithms: The Turnpike Reconstruction Problem, Games

8 Hours

UNIT IV
Algorithm Design Techniques Primality Testing, Backtracking Algorithms: The Turnpike Reconstruction Problem. **Amortized Analysis** Binomial Queues, Skew Heaps. **Advanced Data Structures and Implementation** Top-down Splay Trees, Red-Black Trees: Bottom-Up Insertion.
9Hours

UNIT V
Advanced Data Structures and Implementation Red-Black Trees: Top-Down Red-Black Trees, Top-Down Deletion, Deterministic Skip Lists, AA-Trees, Treaps**,** *k*-d Trees, Merkle trees.
8 Hours

TEXT BOOK

1	Mark Allen Weiss	Data Structures and Algorithm Analysis in C++: Pearson Education, Inc Fourth Edition 2014.
2.	https://www.baeldung.com/cs/merkle-trees	

REFERENCE BOOK

1	Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein	Introduction to Algorithms. Fourth Edition. PHI. 2022.

Course Outcomes: Upon completion of this course the student will be able to:	
CO1.	**Identify** several Abstract Data Types (ADTs) to **implement** various data structures like Binary Trees and Splay Trees and **Apply** the knowledge of Sets and Maps to **design** simple real world problems.
CO2.	**Analyze**, **Design** and **implement** different types of Heap Structures like Binary Heap, d-Heap, Leftist Heap etc.
CO3.	**Analyze** the performance of different sorting algorithms, **apply** algorithm design techniques like randomized and backtracking algorithms to solve problems.
CO4.	**Apply** Amortized Analysis and Compute the complexity of different Heap and Tree data structures.
CO5.	**Implement** advanced data structures Red-Black Trees, Splay Trees and perform various operations on it

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes (PSOs)

		POs												PSOs		
		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
COs	CO1	3	2	1										3		
	CO2		3	2										3		
	CO3		3											3		
	CO4		2											2		
	CO5			2										2		
		3	3	2										3		

Software Defined Networks

Contact Hours/ Week:	**3**	Credits:	3
Total Lecture Hours:	42	CIE Marks:	50
Total Tutorial Hours:	-	SEE Marks:	50
Sub Code:	S7ISPEC033		

Course Objectives:
This course will enable students to:

1. Understand the concepts of traditional networks with their limitations and move to Software defined networks.
2. Understand the fundamentals of SDN, its planar architecture and the respective protocols involved in it.
3. Analyze and evaluate the concept of NFV and their importance in advanced communication
4. Understand the emerging SDN Models and apply the knowledge to design future network problems.

UNIT I

Networking Basics:
Switching, Routing, Addressing SDN Background and Motivation: Evolving network requirements- The SDN approach: Requirements, SDN Architecture and Characteristics of SDN.

8 Hours

UNIT II

SDN Data Plane and OPENFLOW:
Data plane functions, Data plane protocols, OpenFlow: Switch Controller Interaction, Flow Table, Packet Matching, Actions and Packet Forwarding Flow Table Structure, Flow Table Pipeline, The Use of Multiple Tables, Group Table, Extensions and Limitations, Data Plane Scalability

8 Hours

UNIT III

SDN Control Plane
SDN Control Plane Architecture: Control Plane functions, Southbound Interface, Northbound Interface, Routing, Cooperation and Coordination among Controllers, Controller placement Problem, SDN Controllers; OpenDaylight, Ryu, ONOS, Floodlight, Control plane scalability, fault tolerance

SDN Application Plane
SDN Application Plane Architecture: Northbound Interface, Network Applications, User Interface-Network services Abstraction Layer: Abstractions in SDN, Frenetic Traffic Engineering Measurement and Monitoring, Security, Network updates, SDN Usecases.

9 Hours

UNIT IV

Network Function Virtualization
Background and Motivation for NFV- NFV Principles, High level Creation and Chaining, NFV Orchestration, VNF Deployment, Service function Chain Deployment.
NFV Reference Architecture: NFV Management and Orchestration.

9 Hours

UNIT V
Emerging SDN Models: Protocol Models: NETCONF, BGP, MPLS, Controller Models, Application Models: Proactive, Declarative, External, SDN in Datacenters: Multitenancy, Failure Recovery, SDN in Internet eXchange Points (IXPs)
8 Hours

TEXT BOOKS

1	Paul Goransson Chuck Black Timothy Culver	Software Defined Networks: A Comprehensive Approach, Morgan Kaufmann, 2016
2	Ken Gray Thomas Nadeau	Network Function Virtualization, Morgan Kaufmann, 2016

REFERENCE BOOK

1	Larry Peterson, Carmelo Cascone, Bruce Davie	Software Defined Networks: A Systems Approach, 2021
2	Coker, Oswald and Siamak Azodolmolky	Software defined Networking with OpenFlow: Deliver Innovative Business Solutions, Packt Publishing Ltd, 2017

Course Outcomes: Upon completion of this course the student will be able to:	
CO1.	Differentiate between traditional networks and software defined networks and understand the key benefits and use cases of SDN.
CO2.	Interpret the SDN data plane services and OpenFlow Protocols
CO3.	Implement the operation of SDN control plane with different controllers.
CO4.	Apply techniques that enable applications to control the underlying network using SDN.
CO5.	Evaluate Network Functions Virtualization components and their roles in SDN.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes (PSOs)

		POs											PSOs		
		1	2	3	4	5	6	7	8	9	10	11	1	2	3
COs	CO1	2	3										3		
	CO2		3										3		
	CO3		2										3		
	CO4		3										2		
	CO5	3											2		
AVG		3	3										3		

Agile Software Technology and DevOps

Contact Hours/Week	:	3	Credits	:	3.0
Total Lecture Hours	:	42	CIE Marks	:	50
Total Tutorial Hours	:	00	SEE Marks	:	50
Course Code	:	S7ISPEC034			

Course objectives:	
This course will enable students to:	
	Understand traditional testing activities versus agile methodologies.
	Apply agile development practices, design achieve agility.
	Understand and apply key DevOps principles and practices
	Understand DevOps fundamentals, tools, lifecycle, testing, and implementation challenges.

UNIT I
Chapter 1-Introduction What Is Agile Testing, anyway? Agile Values What Do We Mean by Agile Testing? A Little Context for Roles and Activities on an Agile Team, How Is Agile Testing Different, Whole-Team Approach? **Chapter 2 -Ten principles for Agile Testers** What's an Agile Tester? The Agile Testing Mind-Set x CONTENTS, Applying Agile Principles and Values, Adding value.
8 Hours

UNIT II
Chapter 3-Organiational challenges Cultural Challenges, Organizational Culture, Barriers to Successful Agile Adoption by Test/QA Teams, Introducing Change, Management Expectations, Change Doesn't Come Easy. **Chapter 4-Transitioning Typical processes** Seeking Lightweight Processes, Metrics, Defect Tracking, Test Planning, Existing Processes and Models.
8 Hours

UNIT III
Chapter 5- The Agile Testing Quadrants The Agile Testing Quadrants **Chapter 6-Technology- facing Tests that support the team** An Agile Testing Foundation, Why Write and Execute These Tests? Where Do Technology-Facing Tests Stop? What If the Team Doesn't Do These Tests? **Chapter 7- Business- Facing tests that support the team** Driving Development with Business-Facing Tests, The Requirements Quandary, Thin Slices, Small Chunks, How Do We Know We're Done?
9 Hours

UNIT IV
DevOps What is DevOps? DevOps Importance and Benefits, DevOps Principles and Practices 7 C's of DevOps Lifecycle for Business Agility, Continuous Business Planning, Continuous Development, Continuous Integration, Continuous Deployment, Continuous Testing, Continuous Delivery and Monitoring, Continuous Feedback, DevOps and Continuous Testing, Choose Right DevOps Tools Challenges with DevOps implementation.
9 Hours

UNIT V
Map My App to DevOps Journey: Assessment, Definition, Implementation, Measure and Feedback. **Tool Suits**: **Atlassian**: Competitors of Atlassian, Where can Atlassian be Best Utilized, Pros and Cons of Atlassian, Example of Reference Architecture, **Phabricator:** Key Features, Competitors, Where can Phabricator be Best Utilized, Pros and Cons of Phabricator, Example of Reference Architecture.
8 Hours

TEXT BOOKS		
1	Lisa Crispin, Janet Gregory,	Agile Testing: Practical guide for Testers and Agile team, Copyright ©2009 Pearson Education, 2020.
2	Deepak Gaikwad, Viral Thakkar	DevOps Tools from Practitioner's Viewpoint, Wiley, 2019.

REFERENCE BOOKS		
1	Robert C. Martin	Agile Software Development, Principles, Patterns, and Practices, Prentice Hall; 1st edition, 2014.
2	Craig Larman,	Agile and Iterative Development A Manger's Guide, Pearson Education, First Edition, India, 2004.
3	Marc Hornbeek	"Engineering Devops", Kindle edition, 2019, ISBN: 978-0-1234-6578-8

Course Outcomes: Upon completion of this course the student will be able to:	
CO1	**Analyze** the various principles of agile Testers and purpose of agile testing
CO2	**Discuss** various Organizational Challenges and Agile Testing Quadrants.
CO3	**Interpret** the business values of adopting agile development.
CO4	**Comprehend** DevOps fundamentals, tools, lifecycle, testing, and implementation challenges.
CO5	**Analyze** application through DevOps journey using tools and performance feedback.

Course Articulation Matrix

		POs											PSOs		
		1	2	3	4	5	6	7	8	9	10	11	1	2	3
COs	CO1	2											2		
	CO2	2											2		
	CO3	2											2		
	CO4	2		2									2		
	CO5	2											2		
		2	2		2										

Quantum Computing

Contact Hours/Week	:	3	Credits	:	3.0
Total Lecture Hours	:	42	CIE Marks	:	50
Total Tutorial Hours	:	00	SEE Marks	:	50
Course Code	:	S7ISPEC035			

Course objectives: This course will enable students to:	
1.	Introduce foundational concepts of quantum computation, including qubits, quantum gates, and basic quantum algorithms.
2.	Establish the mathematical and physical principles underpinning quantum information theory and quantum mechanics.
3.	Bridge classical and quantum computation, focusing on computational models, complexity classes, and problem-solving paradigms.
4.	Develop the ability to design and analyze quantum circuits and algorithms such as quantum Fourier transform and Grover's search.
5.	Understand the physical realization of quantum computers, including noise models, quantum operations, and hardware constraints.

UNIT I
Introduction and overview: Global perspectives, History of quantum computation and quantum, Information, Future directions. Quantum bits, Multiple qubits, Quantum computation: Single qubit gates, Multiple qubit gates, Measurements in bases other than the computational basis, Quantum circuits, Qubit copying circuit?, Example: Bell states, Example: quantum teleportation, Quantum algorithms: Classical computations on a quantum computer, Quantum parallelism, Deutsch's algorithm, The Deutsch–Jozsa algorithm, Quantum algorithms summarized
9 Hours

UNIT II
Quantum information, Quantum information theory: example problems, Quantum information in a wider context, **Introduction to quantum mechanics:** Linear algebra: Bases and linear independence, Linear operators and matrices, The Pauli matrices, , Inner products, Eigenvectors and eigenvalues The postulates of quantum mechanics: State space, Evolution, Quantum measurement, Distinguishing quantum states, Projective measurements
9 Hours

UNIT III
Introduction to computer science: Models for computation, Turing machines, Circuits , The analysis of computational problems, How to quantify computational resources, Computational complexity , Decision problems and the complexity classes **P** and **NP ,** A plethora of complexity classes, Energy and computation, Perspectives on computer science
8 Hours

UNIT IV
Quantum computation: Quantum circuits, Quantum algorithms, Single qubit operations, Controlled operations , Measurement.

The quantum Fourier transform and its applications: The quantum Fourier transform, Phase estimation, Performance and requirements, **Quantum search algorithms:** The quantum search algorithm: The oracle, The procedure, Geometric visualization, Performance
8 Hours

UNIT V
Quantum computers: physical realization: Guiding principles: Conditions for quantum computation, Representation of quantum information, Performance of unitary transformations , Preparation of fiducial initial states , Measurement of output result. **Quantum noise and quantum operations:** Classical noise and Markov processes , Quantum operations, Overview, Environments and quantum operations , Operator-sum representation, Axiomatic approach to quantum operations
8 Hours

TEXT BOOKS		
1	Michael A. Nielsen & Isaac L. Chuang	Quantum Computation and Quantum Information, University Press, Cambridge, 10th Anniversary Edition, 2010,

REFERENCE BOOKS		
1	Chris Bernhardt	Quantum Computing, Bernhardt, Chris, MIT Press, 1st Edition, 2019
2	Phillip Kaye, Raymond Laflamme, Michele Mosca,	An Introduction to Quantum Computing Oxford University Press, 1st Edition, 2007.

Course Outcomes: Upon completion of this course the student will be able to:	
CO1	Explain the evolution and principles of quantum computation, and implement basic quantum circuits and algorithms.
CO2	Apply quantum mechanics and linear algebra principles to analyze quantum states, operators, and measurements.
CO3	Differentiate between classical and quantum computation models and interpret key computational complexity classes.
CO4	Design and implement advanced quantum algorithms including QFT, phase estimation, and Grover's search.
CO5	Evaluate the physical challenges in implementing quantum computers and explain quantum noise models and quantum operation frameworks.

Course Articulation Matrix

		POs											PSOs		
		1	2	3	4	5	6	7	8	9	10	11	1	2	3
COs	CO1	3	2										2		
	CO2	3	2		2								2		
	CO3	3	2		2								2		
	CO4	3	2	2	2	1							2		
	CO5	3	2		2	2							2		
		3	2		2	2							2		