Semester VII

w.e.f. A.Y. 2021-22

Sr. No.	Course Code	Course Title	Teaching Scheme (Hours/Week)				Examination Scheme				
			L	T	LPW/ PW	C	Duration (Hours)		Component Weightage		
							SEE	LPW/ PW	CE	LPW/ PW	SEE
1	2CS701	Compiler Construction	3	-	2	4	3	2	0.4	0.2	0.4
2	2CS702	Big Data Analytics	2	-	2	3	3	2	0.4	0.2	0.4
3	2CSDEXX	Department Elective –IV	3	-	2	4	3	2	0.4	0.2	0.4
4	2CSDEXX	Department Elective –V	2	-	2	3	3	2	0.4	0.2	0.4
5	2CSOEXX	Open Elective-III	3	-	-	3	3	-	0.6	-	0.4
6	2CSOEXX	Open Elective-IV	3	-	-	3	3	-	0.6	-	0.4
7	2CSXXXX	Minor Project	-	-	4	2	-	2	-	1.0	-
8	2CSXXXX	Summer Internship	-	-	-	1	-	-	-	1.0	-
		Total	16	-	12	23					

L=Lecture, T=Tutorial, P=LPW, C=Credit, SEE=Semester-End Exam, LPW= Lab/Project Work, CE=Continuous Evaluation

Department Elective-IV

S. No.	Course Code	Subjects	L	T	P	C
1	2CSDE76	Mobile Operating Systems	3	-	2	4
2	2CSDE77	Microservice Architecture and Programming	3	-	2	4
3	2CSDE78	Digital Image Processing and Analysis	3	-	2	4
4	2CSDE79	Cloud Security and frameworks	3	-	2	4
5	2CSDE80	Software Testing and Quality Assurance	3	-	2	4
6	2CSDE83	Modern Networks	3	-	2	4
7	2CSDE85	Artificial Intelligence	3	-	2	4

Department Elective-V

S. No.	Course Code	Subjects	L	T	P	C
1	2CSDE86	Application Development Frameworks	2	-	2	3
2	2CSDE87	Ethical Hacking and Vulnerability assessment	2	-	2	3
3	2CSDE88	Simulation and Mathematical Modelling	2	-	2	3
4	2CSDE90	Formal Methods in Software Engineering	2	-	2	3
5	2CSDE93	Blockchain Technology	2	-	2	3
6	2CSDE96	Interfacing with Microprocessors	2	-	2	3

NIRMA UNIVERSITY
Institute of Technology
B.Tech. Computer Science and Engineering
Semester-VII

L	T	P	C
3	0	2	4

Course Code	2CSXXXX
Course Title	Compiler Construction

Course Outcomes:

At the end of the course, students will be able to –

1. summarize the functionalities of various phases of compiler
2. apply language theory concepts to various phases of compiler design
3. identify appropriate optimization technique for compilation process
4. develop a miniature compiler using appropriate compiler design tool

Syllabus	**Teaching Hours: 45**
Unit I **Introduction:** Overview of the Translation Process, Structure of a compiler, Types of compiler and applications, Symbol table	**03**
Unit II **Lexical Analysis:** The role of a Lexical Analyzer, Input Buffering, Specifications of Tokens, Recognition of tokens, Lexical Analyzer Generator, Finite Automata, Regular Expression to Automata, Optimization of DFA based Pattern Matching	**06**
Unit III **Syntax Analysis: C**ontext Free Grammar, Top-down Parsing, Bottom-up Parsing, LR Parsers, Error Recovery, Parsing for ambiguous grammars, Parsing Generator Tools	**13**
Unit IV **Syntax Directed Translation:** Syntax Directed Definition (SDD), Evaluation order of SDD, Syntax Directed Translation Schemes	**07**
Unit V **Intermediate Code Generation:** Variants of Syntax Trees, Three Address Codes, Type Checking, Control Flow, Back patching	**07**

Unit VI **02**

Runtime Environment: Storage Organisation, Stack Allocation and Heap Management

Unit VII **07**

Code Generation and Optimization: Issues in code generation, Data Flow and Control Flow, Peephole Optimization, Register Allocation, Machine independent optimization techniques **Self-Study:**

Compiler Generation tools like LEX, YACC etc

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Aho, Lam, Ullman, Sethi, Compilers, Principles, Techniques and Tools, Pearson
2. Keith D Cooper & Linda Torczon, Engineering a Compiler, Elsevier
3. Jean Paul Trembly & Paul G Sorenson, The theory and Practice of Compiler writing, McGraw Hill

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list

Suggested List of Experiments:

Sr. No	Title	Hours
1	To implement lexical analyse to recognize all distinct token classes.	02
2	To implement a Recursive Descent Parser Algorithm for the grammar.	02
3	To find the First () and Follow () of a grammar.	04
4	To implement the Left most derivation removal algorithm.	04
5	To implement a calculator in YACC.	04
6	To generate Three Address code for assignment statement	02
7	To implement grammar rules for control statements, and Loop control.	04
8	To implement a Type Checker.	02
9	To implement Assembly code generator.	02
10	To implement Code Optimization techniques.	04

NIRMA UNIVERSITY
Institute of Technology
B. Tech. Computer Science and Engineering
Semester-VII

L	T	P	C
2	0	2	3

Course Code	2CSXXXX
Course Title	Big Data Analytics

Course Outcomes:

At the end of the course, students will be able to -

1. outline the significance and challenges of big data
2. model big data using different tools and frameworks
3. apply big data techniques for useful business analytic applications
4. design algorithms for mining the data from large volumes

Syllabus

Teaching Hours: 30

Unit I

Introduction to Big Data: Evolution of Big Data, Types of Digital Data, Classification of Digital Data, Structured Data, Semi-Structured Data, Unstructured Data, Definition of Big Data, Challenges of Conventional Systems, Big data platforms and data storage — **04**

Unit II

Big Data Analytics: Importance of Big data analytics, Classification of Analytics, Top Challenges Facing Big Data, Technologies to meet the Challenges Posed by Big Data, Terminologies Used in Big Data Environment — **04**

Unit III

Hadoop: Introducing Hadoop, comparisons of RDBMS and Hadoop, Distributed Computing Challenges, Hadoop Overview, Business Value of Hadoop, Hadoop Distributed File System, Processing Data with Hadoop, working with Map Reduce, Hadoop YARN, Hadoop in the Cloud, Applications on Big Hadoop — **08**

Ecosystem, Fundamentals of Pig, Hive, HBase and ZooKeeper, Basic concepts of Apache Spark

Unit IV

The Big data technology landscape: CAP Theorem - BASE Concept, NoSQL, Types of No SQL databases, Introduction to MongoDB, Data Types in MongoDB, CRUD, Apache Cassandra, Features of Cassandra, CRUD **08**

Unit V **06**

Big data analytics Algorithm: Applying Linear Regression, Clustering, Association rule mining, Decision tree on Big Data. **Self-Study:**

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer
2. Tom White, Hadoop: The Definitive Guide, O'reilly Media
3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw Hill Publishing
4. Anand Rajaraman and Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press
5. Bill Franks, Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, John Wiley & sons
6. Glenn J. Myatt, Making Sense of Data, John Wiley & Sons
7. Da Ruan, Guoquing Chen, Etienne E.Kerre, GeertWets, Intelligent Data Mining, Springer
8. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, Harness the Power of Big Data the IBM Big Data Platform, Tata McGraw Hill Publications
9. Michael Minelli, Michele Chambers, Ambiga Dhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Wiley Publications
10. Zikopoulos, Paul, Chris Eaton, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, Tata McGraw Hill Publications
11. Seema Acharya and Subhashini C, Big Data and Analytics, Wiley India

L=Lecture, T=Tutorial, P=Practical, C=Credit

^this is not an exhaustive list

Suggested List of Experiments:

Sr. NO	Practical Title	Hours
1	Learning limitation of data analytics by applying Machine Learning Techniques on large amount of data. Write R/Python program to Read data set from any online website, excel file and CSV file and to perform a) Linear regression and logistic regression on iris dataset. b) K-means clustering.	02
2	Setup single node Hadoop cluster and apply HDFS commands on single node Hadoop Cluster. (*students can setup multimode cluster in laboratory)	04
3.	Apply MapReduce algorithms to perform analytics on single node cluster: a) Analyze phrase frequency from given dataset b) Search Records with matching criteria c) Aggregate inputs and search records based on aggregation	04
4	Analyze impact of different number of mapper and reducer on same definition as practical 3.	02
5	Implement PCY/Multi-Hash/SON algorithm for identification of frequent item set by handling larger datasets in main memory.	02
6	Setup the MongoDB environment in your system. Import Restaurant Dataset and perform CRUD operation.	02
7	Extend MongoDB functionality for MapReduce on document collection	02

8	SPark SQL and MLLib: (iii) PYspark shell exploration and reading and writing in HDFS (iv) Clustering using MLlib , compare results of clustering with Hadoop MR and with Spark	02
9	Identify a case study to perform analytics on different platforms (like NoSQLs, Spark, Zookeeper and analyse differences.	04
10	Case study: Use the following platforms for solving any big data analytic problem of your choice. (1) Amazon web services,(2) Microsoft Azure, (3)Google App engine	02

NIRMA UNIVERSITY
Institute of Technology
B. Tech. Computer Science and Engineering
Semester-VII
Department Elective-IV

L	T	P	C
3	0	2	4

Course Code	2CSDEXX
Course Title	Mobile Operating Systems

Course Outcomes:

At the end of the course, students will be able to -

1. compare the similarities, differences and benefits of the current mobile operating systems
2. explain the functionalities of remote operations and security essential of mobile devices
3. analyze the latest trends in building Mobile OS
4. demonstrate the native applications required to build using mobile OS

Syllabus Teaching Hours: 45

Unit I **10**

Introduction: Introduction to different operating system platforms, Windows Server, Windows Desktop, Linux servers, Android, Apple iOS.

Unit II **08**

List and order the basic process of building a PC, component selection, order of component assembly, environment precautions.

Describe the process for installing a software operating system, obtaining installation media, identifying suitable hardware, installing software, configuring for first use.

Unit III **10**

List and describe the order of tasks required for end-to-end testing of an operating

system to ensure it works as intended (Windows, Linux), perform a log in as an administrative user, test remote management, perform a log in as a normal user, verify that a normal user cannot use admin tools requiring elevated permissions, verify that connectivity to network resources and internet services works correctly.

Unit IV **09**

Summarise the native applications for different operating systems. IOS, Safari, Maps, App Store, Windows; o IE, Edge, Notepad, Paint, Command Prompt, Linux, Nano, Terminal, Android; Chrome, Maps, Play Store.

Unit V **08**

Explain the security principles when running an operating system running on a platform; with a focus on physical hardware, virtual servers and cloud services. secure configuration following recommended good practice; user access control; malware protection; patch management.

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Gerardus Blokdyk, Mobile Operating System A Complete Guide, 5STARCooks publication
2. Reto Meier, Professional Android 4 Application Development, Wrox Publication
3. Books Llc (Google online Books), Mobile Phone Operating System : General Books LLC 4. Haseman, Chris, Android Essentials Apress publication
5. Charlie Miller, Dion Blazakis, Dino DaiZovi, Stefan Esser, Vincenzo Iozzo, iOS Hacker's Handbook, Wiley publication

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list

Suggested list of Experiments:

Sr. No	Title	Hours
1	To explore different operating system platforms	04
2	To perform the process for installing a software operating system	02
3	To explore and implement the order of tasks required for end-to-end testing of an operating system to ensure it works as intended (Windows, Linux)	02
4	To study and perform algorithmic verification (model-checking) of operating system	04
5	To summarise the native applications for different operating systems	04
6	To implement the security concepts when running an operating system running on a platform; with a focus on physical hardware, virtual servers and cloud services	04
7	Mobile application phase-I: Demonstration of simple UI with user activity To develop an application by using list view and spinner view to list out some items on screen by selecting any of the items system display complete information about that item. Also show addition and deletion of the items from the spinner view.	02
8	Mobile application phase-II: To show Communication between two Activity through Intents: In first Activity take two input from user send these two numbers or strings via intent to second activity, perform the user defined operations on these two numbers/strings in second activity. Now sent back to first activity and show the results to the user on first activity.	04
9	Study and explore Internet of Things: To design a program to access sensors and control actuator using HTTP protocol. Connect ESP8266 to a WiFi network and control/access the sensor/actuator from web browser using HTTP.	02
10	To study and explore different Remote Systems Management API.	02

NIRMA UNIVERSITY
Institute of Technology
B. Tech. Computer Science and Engineering
Semester-VII
Department Elective-IV

L	T	P	C
3	0	2	4

Course Code	2CSDEXX
Course Title	Microservice Architecture and Programming

Course Outcomes:

At the end of the course, students will be able to -

1. recognize the key advantages and complexities present in microservice architectures
2. apply appropriate architectural approach for the design of microservices
3. implement microservice applications effectively with the suitable techniques and technologies
4. test the deployment of microservice applications on cloud platforms

Syllabus

Teaching Hours: 45

Unit I **03**

Introduction to Microservices: Monolithic architecture, Web Services and

Service Oriented Architecture, SOA and Microservice architecture

Unit II **08**

Microservice Architecture Concepts: Microservice software architecture: patterns and techniques, Overall topology and core architecture components, Architectural characteristics, Service components and granularity, Bounded context, Data domains, API Ecosystem for Microservice, API layer design and implementation alternatives, API Gateway, Service discovery and registration, Best practices of microservice architecture

Unit III **10**

Messaging Middleware: IPC in microservice architecture, Synchronous and asynchronous messaging patterns, REST and gRPC based messaging, Service bus for commands and events, Message queuing systems, Message broker, JMS, Rabbit MQ and Kafka, Message driven micro service application

Unit IV **06**

Managing Databases for Microservices: Distributed databases, NoSQL based systems, CAP and BASE consistency models for microservices, CRUD operations, Shared databases and Database per microservice pattern, Scaling and replicating databases

Unit V **08**

Transactions and Data Streaming in Microservices: Managing transactions with Sagas: choreographed, orchestrated, Event sourcing and CQRS Pattern, CDC with Transactional outbox pattern, Transaction log tailing, Streaming data in microservices, Streaming SQL, Data streaming approaches with Apache Spark and Kafka

Unit VI **10**

Hybrid Architectures and Deployment: Event-driven architecture for microservices, Architectural modularity, Serverless microservices architecture pattern, Caching, Load balancing, Circuit Breaker, Deployment patterns and strategies with containers, Virtual machines and clusters, Container Orchestration Approaches, Mircoservices deployment on Public Cloud platforms, Microservices Testing, Healthcheck and observability, Securing Microservices

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Chris Richardson, Microservices Patterns With examples in Java, Manning publication
2. Binildas C, Practical Microservices Architectural Patterns, Apress
3. Sam Newman, Building Microservices: Designing fine grained systems, O'Reilly Media
4. Sam Newman, Monolith To Microservices, O'Reilly
5. Irakli Nadareishvili, Ronnie Mitra, Matt McLarty, Mike Amundsen, Microservice Architecture: Aligning Principles, Practices, and Culture, Shroff/O'Reilly
6. Susan J. Fowler, Microservices in Production, O'Reilly Media
7. Morgan Bruce, Paulo A. Pereira, Microservices in Action, Manning publication
8. Vaughn Vernon, Implementing Domain-Driven Design, Addison-Wesley
9. Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra, Head First Design Patterns: A Brain-Friendly Guide, Shroff/O'Reilly
10. Jez Humble and David Farley, Continuous Delivery, Addison-Wesley Professional
11. Bill Wagner, Mike Rousos, .NET Microservices: Architecture for containerized .Net applications, Microsoft Corporation

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list

Suggested List of Experiments:

Sr. No.	Title	Hours
1	Experimenting with Containers and understanding its fundamentals with basic operations on it	04
2	Cloud Native Application Development Language: Hands-on Sessions for basic concepts learning	06
3	Designing gRPC based Micro-service application	04
4	Message queuing system based Micro-service application development	04
5	Designing Distributed transaction (Saga) based Micro-service application	04
6	Integrating various Micro-services with application of API Gateway	04
7	Scalable and Resilient Micro-service application design with security provisions for the services	04

NIRMA UNIVERSITY
Institute of Technology
B. Tech. Computer Science and Engineering
Semester-VII
Department Elective-IV

L	T	P	C
3	0	2	4

Course Code	2CSDEXX
Course Title	Digital Image Processing and Analysis

Course Outcomes:

At the end of the course, students will be able to –

1. illustrate basic image acquisition mechanisms and image formats
2. identify various applications of digital image processing techniques
3. apply mathematical principles and signal processing concepts in digital image enhancement and restoration
4. develop various image representation stages for digital image processing applications

Syllabus:

Total Hours: 45

Unit I **05**

Introduction: Origin and Applications of subject, Fundamental Steps in Digital Image Processing and Components of an Image Processing System, Image Sampling and Quantization, Basic Relationships between Pixels.

Unit II **08**

Image Enhancement in the Spatial and Frequency Domain: Background, Some Basic Gray Level Transformations, Histogram Processing Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, Introduction to the Fourier Transform and the Frequency Domain, Smoothing and Sharpening Frequency Domain Filters

Unit III **10**

Image Restoration: A Model of the Image Degradation/Restoration Process, Noise

Models, Restoration in the Presence of Noise Only- Spatial Filtering, Periodic Noise

Reduction by Frequency Domain Filtering, Linear, Position-invariant Degradations,

Estimating the Degradation Function, inverse Filtering.

Unit IV **10**

Color Image Processing : Pseudo color Image Processing, Color Models**.**

Image Compression: Fundamentals, Image Compression codecs (JPEG, PNG, GIF)

Morphological Image Processing: Erosion, Dilation, Opening, Closing, Hit or Miss Transformation, Boundary Extraction, Hole Filling, Extraction of Connected Components, Convex-Hull, Thinning, Thickening, Skeletons, Pruning.

Unit V **07**

Image Segmentation: Edge Models Discontinuity based Image Segmentation, Similarity Based Image Segmentation

Representation and Description: Representation, Boundary Descriptors, Regional Descriptors

Image denoising: Challenges and Techniques

Unit VI **05**

Case Study: Geo Spatial Image processing, Medical Image processing, thermal Image processing, sonar Image processing

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson education
2. Anil K. Jain, Fundamental of Digital Image Processing, Prentice Hall
3. Kenneth R. Castleman, Digital Image Processing, Paperback
4. Earl Gose, Johnsonbaug, Steve Jost, Pattern Recognition and Image Analysis PHI.
5. Rafael C. Gonzalez, Image Processing using MATLAB, PHI

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list

Suggested List of Experiments:

Sr. No.	Title	Hours
1.	To write a computer program capable of reducing the number of intensity levels in an image from 256 to 2, in integer powers of 2. The desired number of intensity levels needs to be a variable input	02
2.	To implement zooming and shrinking of an image using bilinear interpolation.	02
3.	To write a program to load an image and then perform a simple spatial 3x3 average of image pixels. Repeat the process for a 5x5 neighbourhood and again for a 7x7 neighbourhood.	04
4.	To perform intensity transformation on a given image a. log transformation b. gamma transformation c. contrast stretching	04
5.	To experiment and analyse, how histogram equalization will help enhance the image provided during the lab session. Also use equalization on 5 different types of images and find out in which case it gives the weakest enhancement.	02
6.	To write a program that performs image enhancement using spatial domain filters.	02
7.	Write a programme: a. that performs image enhancement using frequency domain filters b. that accepts a noisy image and performs restoration.	04
8.	To write a program that detects the following on an appropriate input image a. Line and edge detection using mask filters. b. Discontinuity detection	04
9.	To segment the grey scale image and finding appropriate representation for the segmented shape	02
10.	To write a program that provides an Interactive Segmentation for selection of object of interest and apply the user specified effect on the background. (Minimum offered Effects: Blur, Pencil Sketch, Water Colour, Vibrant, black and white, black and white red filter, pseudo color, canvas)	04

NIRMA UNIVERSITY
Institute of Technology
B. Tech. Computer Science and Engineering
Semester-VII
Department Elective-IV

L	T	P	C
3	0	2	4

Course Code	2CSDEXX
Course Title	Cloud security and frameworks

Course Outcomes:

At the end of the course, students will be able to -

1. classify cloud architectural aspects
2. recognize the trusted platform for cloud computing.
3. identify the security risks associated with the cloud platforms
4. inspect the cloud computing security design patterns

Syllabus

Teaching Hours: 45

Unit I **10**

Fundamentals of cloud computing architecture and security issues: Current Cloud standards, various protocols used, and best practices intended for delivering Cloud-based services. Identification of the cloud threats, vulnerabilities, and privacy issues of cloud-based IT services. Common attack vectors and threats, encryption, data redaction, tokenization, obfuscation, PKI and Key management, assuring data deletion, data retention, and archiving procedures for tenant data, data protection strategies.

Unit II **09**

Cloud Design principles with the perspective of security: Identification and understanding of the core principles for designing and implementing appropriate safeguards and methods for cloud services. Secure isolation strategies multitenancy, virtualization strategies inter-tenant network segmentation strategies, storage isolation strategies. OS Hardening and minimization, securing remote access, verified and measured boot, firewalls, IDS, IPS, and honeypots. end-to-end identity and access management, monitoring and auditing processes, and compliance with industry and regulatory mandates.

Unit III **09**

Trusted Platform for Cloud: Trust and Reputation Model in Cloud, trusted cloud resource pools, secure cloud interfaces, cloud data breach protection, permanent data loss protection, cloud traffic hijacking protection, cloud authentication gateway, federated cloud authentication cloud key management, trust attestation service, collaborative monitoring and logging independent cloud auditing. Privacy requirements for Cloud computing, metrics for service level agreements (SLA), metrics for risk management. Study of Docker and cloud agnostic architecture and its security issues.

Unit IV **08 Cloud Computing Security Design Patterns-I**: Security Patterns for Cloud Computing, Geo-tagging, Cloud VM Platform Encryption, Cloud Resource Access Control, Cloud Data Breach Protection, Permanent Data Loss Protection, In-Transit Cloud Data Encryption.

Unit V **09**

Cloud Computing Security Design Patterns –II: Security Patterns for Cloud Computing, network security, identity & access management & trust secure onpremise internet access, secure external cloud connection, cloud denial-of-service protection, cloud traffic hijacking protection, automatically defined perimeter, cloud authentication gateway federated cloud authentication, cloud key management, Case Study: HIPAA compliance.

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with a minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Vic (J.R.) Winkler, Securing The Cloud: Cloud Computing Security Techniques and Tactics, (Syngress/Elsevier).
2. Thomas Erl, Cloud Computing Design Patterns, Prentice Hall.
3. Raj Samani, Brain Hoanan, Jem Reavis, Vladimir Jirasek, CSA guide to Cloud Computing: Implementing Cloud Privacy and Security, Elsevier.
4. Chris Doston, Practical Cloud Security: A guide for secure design and deployment, OREILLY.

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list

Suggested List of Experiments:

Sr. No.	Title	Hours
1	To work with AWS IAM (Identity Access Management) to assign the various rights to the cloud user for dedicated services	02
2	To configure the private cloud for performing the security approaches and creating a Test bed to perform various attacks and identifying its effect	04
3	To introduce the Command line configuration for open stack open source cloud.	02
4	To get familiar with End-User/ Cloud Operator operations to be conducted in the openstack environment.	02
5	To understand the network topology and its configuration for open stack open source cloud.	04
6	To explore and implement the open-source cloud security tools. To understand and analyse its impact to the cloud resources components	02
7	To perform a DDoS simulation attack and identifying its pattern using Wireshark tool/ or any other networking tool (Goldeneye simulator)	02
8	To implement the cloud monitoring strategy on any public/ private cloud and identify the traces of the attack	04
9	To identify the SLA violation using Rally and analysing its results in terms of a graph representation and to trace the anomaly detection.	04
10	To perform the malware analysis using a suspicious hash repository from virus total API.	04
11*	Introduction to libVMI for virtual machine monitoring using VM inspection tool.	04

NIRMA UNIVERSITY
Institute of Technology
B.Tech. Computer Science and Engineering
Semester-VII
Department Elective-IV

L	T	P	C
3	0	2	4

Course Code	2CSDEXX
Course Title	Software Testing and Quality Assurance

Course Outcomes:

At the end of the course, students will be able to -

1. interpret different types of testing techniques in depth
2. apply modern software testing strategies in relation to software development
3. design project test plans, test cases, test data to conduct test operations
4. develop practical skills related to software quality assurance

Syllabus

Teaching Hours: 45

Unit I **12**

Overview of Software Testing: Software Quality, Role of testing, testing approaches

Unit Testing: Concept of Unit Testing, Defect Prevention, Mutation Testing, Debugging, Unit Testing in eXtreme Programming

Control Flow Testing: Control Flow Graph, Paths in a Control Flow Graph, All-Path Coverage Criterion, Statement Coverage Criterion, Branch Coverage Criterion, Examples of Test Data Selection

Data Flow Testing: Data Flow Anomaly, Data Flow Graph, Data Flow Testing Criteria, Feasible Paths and Test Selection Criteria, Comparison of Testing Techniques.

Unit II **08**

System Integration Testing: System Integration Techniques, Types of Interfaces and Interface errors, Software and Hardware Integration, Off the shelf component testing, Built-in Testing

System Test Categories: Basic Tests, Functionality Tests, Robustness Tests, Interoperability Tests, Performance Tests, Scalability Tests, Stress Tests, Load and Stability Tests, Reliability Tests, Regression Tests, Documentation Tests

Functional Testing: Equivalence Class Partitioning Boundary value Analysis, Decision Tables, Random Testing, Error guessing, Category Partition

Unit III **08**

System Test Design and Planning: Test Design Factors, Requirement Identification, Characteristics of Testable Requirements, Test Design Preparedness Metrics, Test Case Design Effectiveness

System Test Planning, Automation: Structure of a System Test Plan, Beta Testing, System Test Automation

Unit IV **07**

System Test Execution: Metrics for tracking System Test, Beta Testing, System Test Report, Product Sustaining, Measuring Test Effectiveness

Acceptance Testing: Types of Acceptance Testing, Selection of Acceptance Criteria, Acceptance Test Execution, Acceptance Testing in eXtreme Programming.

Use cases for State-of-the-Art tools for carrying out Software Testing Unit V **10**

Software Quality Assurance: Five views of Software Quality, McCall's Software Quality Factors, Quality Criteria, Relationship between Quality factors and Criteria, Components of SQA, Software Quality Standards and their requirements, Software Quality Metrics, Software Reliability Models

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Sagar Naik, Piyu Tripathy, Software Testing and Quality Assurance: Theory and Practice, Wiley.
2. William Perry, Effective Methods for Software Testing, Wiley.
3. Paul C. Jorgensen, Software Testing - A Craftsman's Approach, CRC Press.
4. Srinivasan Desikan and Gopalaswamy Ramesh, Software Testing, Pearson Education.
5. Louis Tamres, Introducing to Software Testing, Addison Wesley Publications.
6. Ron Patton, SAMS Techmedia Indian Edition, Software Testing, Pearson Education.
7. Glenford J. Myers, The Art of Software Testing, John Wiley & Sons.
8. Robert V. Binder, Testing Object-Oriented Systems: Models Patterns and Tools, Addison Wesley.
9. Daniel Galin, Software Quality Assurance: From Theory to Implementation, Addison Wesley.
10. Stephen Kan, Metrics and Models in Software Quality Engineering, Addison Wesley.

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list.

Suggested list of Experiments:

Sr. No.	Title	Hours
1	To analyse given webpages from user interface, functionality, and security perspective, and perform manual testing	02
2	To perform comparative analysis of software testing tools	02
3	To study and perform sample tests using Test Link Testing tool	04
4	To study and perform sample tests using J-Unit Testing tool	04
5	To study and perform sample tests using Selenium Testing tool	04
6	To perform automated functional GUI testing using any tool	04
7	To demonstrate performance-based testing experiment	02
8	To implement load testing method	02
9	To demonstrate regression-based testing experiment	04
10	To demonstrate testing of a mobile application	02

NIRMA UNIVERSITY
Institute of Technology
B. Tech. Computer Science and Engineering
Semester-VII
Department Elective-IV

L	T	P	C
3	0	2	4

Course Code	2CSDEXXX
Course Title	Modern Networks

Course Outcomes:

At the end of the course, students will be able to -

1. interpret the modern networking concepts and trends
2. demonstrate basic skills for cellular networks design
3. apply the modern networking fundamentals on real-time network analysis
4. design various types of networks using appropriate tools

Syllabus

Teaching Hours: 45

Unit I **08**

Network Concepts and Congestion Control: Networking Principles, Network elements, Performance of networks, Router architecture and switching fabric in routers, congestion control in network, Analysis of TCP, QoS and fairness, Traffic shaping and TCP flow and congestion control

Unit II **04**

Software Defined Networking: Data Plane, Control Plane, Application Plane, Controller design, Virtualization, OpenFlow protocol for SDN, Network Function Virtualization

Unit III **08**

Data Center Networking: Data center architectures, Data enter congestion control, Queuing and traffic patterns, Data center network protocols, End host architectures, ECMP and load balancing, Multipath TCP, DCTCP, Deadlineaware DCTCP, Low latency protocols for data center

Unit IV **10**

Next Generation Mobile Networks: Basics of cellular networks, GSM, GPRS, 3G, 4G, LTE - architecture and working, 5G architecture and objectives, working principles, Beamforming and hardware technologies for mmW communications, Software Defined radio

Unit V **5**

IPv6 and Lightweight IP Stack: Need of µIP, I/O processing and packet forwarding, Buffer management and API for µIP, Protocols implementations for µIP, IPv6 addressing, Anycast and multicast with IPv6, IPv4 and IPv6 interoperability

Unit VI **10**

Case Studies: Backbone of Internet, Internet exchange points and BGP, Large scale data centers, Peer-to-peer systems, Content Delivery Networks, Multimedia networks, Video streaming networks, Content-centric Networks, Li-Fi, Blockchain Technology, Cognitive radio networks, Bare Metal Networking

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. William Stallings, Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, Addison-Wesley
2. Kurose and Ross, Computer Networking: A Top-Down Approach, Addison-Wesley
3. William Stallings, High-speed networks and Internets – Performance and quality of service, Prentice Hall
4. Huitema, C., Routing in the Internet, Prentice-Hall,
5. Keshav, S., An Engineering Approach to Computer Networking, Addison-Wesley
6. Jean-Philippe Vasseur, Interconnecting Smart Objects with IP, Morgan Kaufman Publisher
7. Asif Oseiran, Jose F.Monserrat and Patrick Marsch, 5G Mobile and Wireless Communications Technology, Cambridge University Press.
8. Martin Sauter, From GSM to LTE-Advanced Pro and 5G: An Introduction to Mobile Networks and Mobile Broadband, John Wiley & Sons Ltd.
9. W. Richard Stevens and Gary R. Wright, The TCP/IP Illustrated, Volume 1: The Protocols, Addison Wesley Longman
10. W. Richard Stevens and Gary R. Wright, "The TCP/IP Illustrated, Volume 2: The Implementation", Addison Wesley Longman.
11. Syed Hassan Ahmed, Safdar Hussain Bouk and Dongkyun Kim, Content-Centric Networks - An Overview, Applications and Research Challenges

1. Naoaki Yamanaka, High-Performance Backbone Network Technology, CRC Press.

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list.

Suggested List of Experiments:

Sr. No.	Title	Hours
1	To configure an IPv4 network using packet tracer and perform static routing and dynamic routing using RIP, RIPv2 and RIPng	04
2	To configure an IPv6 network using packet tracer and perform address assignment to nodes, static routing, dynamic routing and implement DHCP server to automatically assign IPv6 addresses	04
3	To configure Autonomous networks using Border Gateway Protocol in packet tracer.	02
4	To implement a basic TCP Client Server echo application using sockets in C language.	02
5	To define data, control and application planes in software defined networking in Openflow.	04
6	To simulate cellular network for LTE and 5G	04
7	To implement a basic IPv6 μIP service for configuring and testing IPv6 nodes	02
8	To configure video streaming server and stream over network for the client player	04
9	To study various consensus algorithms for blockchain implementation. For the application of your choice, apply appropriate consensus algorithm.	02
10	To demonstrate the use of blockchain for the application of your choice using the consensus algorithm implemented in practical 9.	02

NIRMA UNIVERSITY
Institute of Technology
B. Tech. Computer Science and Engineering
Semester-VII
Department Elective-IV

L	T	P	C
3	0	2	4

Course Code	2CSDEXX
Course Title	Artificial Intelligence

Course Outcomes:

At the end of the course, students will be able to –

1. explain the significance of Artificial Intelligence and knowledge representation,
2. demonstrate the design concepts of control and search strategies in AI Applications, 3. compare different search strategies for a given scenario
4. design applications using Artificial Intelligence.

Syllabus

Teaching Hours: 45

Unit I **05**

Introduction to Artificial Intelligence Overview:

Knowledge: General concepts, definition and importance of knowledge, knowledgebased system, representation, organization, manipulation and acquisition of knowledge.

Unit II **14**

Problems, Problem Spaces and State Space Search: The AI Problems, The Underlying Assumption, What Is an AI Techniques, The Level of The Model, Criteria For Success, Some General References, One Final Word. Defining the Problems as a State Space Search, Production Systems, Production Characteristics, Production System Characteristics, and Issues in The Design of Search Programs.

Search and Control Strategies: Uninformed (Blind) and informed search, DFS, BFS, Heuristic **Search Techniques:** Generate-And-Test, Hill Climbing, Best-First Search, A*, AO*, Problem Reduction, Constraint Satisfaction, Means-Ends Analysis.

Unit III **12**

Knowledge Representation: Knowledge Representation Issues, Representations and Mappings, Approaches to Knowledge Representation, Using Predicate Logic Representation Simple Facts in Logic, Representing Instance and ISA Relationships, Computable Functions and Predicates, Resolution. Representing Knowledge Using Rules, Procedural versus Declarative Knowledge, Logic Programming, Forward Versus Backward Reasoning.

Unit IV **09**

Weak Slot-And-Filler Structure: Semantic Nets, Frames

Reasoning: Symbolic Reasoning under Uncertainty, Introduction to Non-monotonic Reasoning, Logics for Non-monotonic Reasoning. Statistical Reasoning , Probability And Bay's Theorem, Certainty Factors And Rule-Base Systems, Bayesian Networks, Dumpster-Shafer Theory.

Unit V **05**

Game Playing: Overview and Example Domain, Min-max Search, Adding AlphaBeta Cutoffs. Expert System: Introduction, Architecture, and Types of Expert Systems, Expert System shell.

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Russel and Norvig, Artificial Intelligence:A modern approach, prentice Hall
2. Elaine Rich And Kevin Knight, Artificial Intelligence, Tata McGraw-Hill
3. D.W.Patterson, Artificial Intelligence And Expert Systems, Prentice Hall
4. D.W.Rolston , Artificial Intelligence And Expert System Development, Mcgraw-Hill
5. Ivan Bratko, PROLOG Programming for Artificial Intelligence, Addison-Wesley

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list

Suggested List of Experiments:

Sr. No.	Title of the Practical	Hours
	Explore open-source AI tools. Submit write up on AI tools in following form.	02

<table>
<tr><td>1</td><td><table>
<tr><th>Tool Name</th><th>Advantages/Best suitable scenarios</th><th>Applications</th></tr>
<tr><td>AIML</td><td>Natural language processing scenarios</td><td>Chatbots</td></tr>
</table></td><td></td></tr>
<tr><td></td><td>Use C/C++/JAVA/Python for following practical</td><td></td></tr>
<tr><td>2</td><td>Write a program to implement DFS (for 8 puzzle problem)</td><td>02</td></tr>
<tr><td>3</td><td>Write a program to implement BFS (for 8 puzzle problem)</td><td>02</td></tr>
<tr><td>4</td><td>Write a program to implement A * (for 8 puzzle problem)</td><td>04</td></tr>
<tr><td></td><td>Use PROLOG for the following experiments:</td><td></td></tr>
<tr><td>5</td><td>Write a program in PROLOG for Query based on family tree
Knowledge base contain the data of at-least three generations.</td><td>02</td></tr>
<tr><td>6</td><td>Write a PROLOG program on lists lists.
i. To find whether given element is a member of list ii.
Inserting an element at a) beginning b) end c) desired position</td><td>02</td></tr>
<tr><td>7</td><td>Write a PROLOG program on lists.
i. Reversing a list.
ii. Finding the position of given element in the list
a) from beginning.
b) from end.</td><td>02</td></tr>
<tr><td></td><td>Implementation of Checkers/Tic-Tac-Toe Game using the following Algorithms (8 & 9)</td><td></td></tr>
<tr><td>8</td><td>Min-max Algorithm</td><td>04</td></tr>
<tr><td>9</td><td>Alpha-beta pruning</td><td>04</td></tr>
<tr><td>10</td><td>Design an Expert System of your choice.</td><td>02</td></tr>
</table>

NIRMA UNIVERSITY
Institute of Technology
B.Tech. Computer Science and Engineering
Semester-VII
Department Elective-V

L	T	P	C
2	0	2	3

Course Code	2CSDEXX
Course Title	Application Development Frameworks

Course Outcomes:

At the end of the course, students will be able to -

1. interpret basic concepts of application development frameworks
2. explain message framework in Django
3. develop programs to solve real world problems based on concepts of Django
4. design applications using cross platform development concepts

Syllabus

Teaching Hours: 30

Unit I **02**

Introduction: Introduction about different Web and mobile application-based frameworks, Introduction to Django and kivy, The MVT and MVC Design Pattern, Environment Setup, Project Structure, Apps Life Cycle

Unit II **09**

View, Form and Templates in Django: Creating Django Views, URL mapping, basic of templates, template variables, built-in and custom tags and filters, using templates in views, template inheritance, assets handing, Form creation using Django, Form rendering process, build-in and custom widgets, Formsets, generating PDF

Unit III **07**

Django models: Interacting with a Database: Basics, Installation and Configuration of Database, defining model, database CRUD operations, rendering model in Admin interface, Fields validation, file uploads

Unit IV **05**

Advance topics in Django: Customizing Authentication, Admin Interface and its control, User and group creation, permission handling, session and cookies handling, sending mails, message framework

Unit V **07**

Cross-platform application development: Basic of kivy, Creating views, Navigation drawer, Layouts and Screen Manager, APK file generation **Self-Study:**

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. William S Vincent: Django for Beginners: Build websites with Python and Django,
2. Nigel George: Build a Website With Django 3: A complete introduction to Django 3,
3. Antonio Mele: Django 3 By Example: Build powerful and reliable Python web applications from scratch, Packt Publishing Limited
4. Online Django documentation: https://docs.djangoproject.com/en/3.1/
5. William S Vincent: Django for Professionals: Production websites with Python & Django
6. William S Vincent: Django for APIs: Build web APIs with Python and Django
7. Daniel Roy Greenfeld: Two Scoops of Django 1.11: Best Practices for the Django Web Framework, Two Scoops Press
8. Tarkeshwar Barua, Ruchi Doshi, Kamal Kant Hiran: Mobile Applications Development: with Python in Kivy Framework, De Gruyter

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list.

Suggested List of Experiments:

Sr. No.	Title	Hours
1	To explore the Django application structure and design simple hello world webpage	02
2	To design a new social media platform using Django. Design a view, templates and filters for different modules. Ensure that your application has following features: 1. Login and registration page 2. The dashboard which displays the latest post 3. Every post can be liked, shared by others 4. Another user is permitted to comment on displayed post	04
3	To design an admin module with dashboard and grant management privileges like admin can update and delete the content of any user	02

4	To perform CRUD operation and connect database with your social media website. Store all the data and retrieve the data according the user	04
5	To manage session and cookies in your social media platform	02
6	To enhance the admin interface and customize the authentication rules and permission handling module, group handling	04
7	To include an advance feature such as recommendation friend to the user based on common features they are sharing.	02
8	To design a simple Expense management system using kivy, compile it and generate .apk file	04
9	To improve the Expense management system developed in previous practical by implementing navigation drawer and grid layout	02
10	To integrate various visualization packages in python and improve the previously developed Expanse management system. Categories your expenses and visualize them using pie-chart based on daily, weekly, monthly, quarterly and yearly data. Give some generalized tips to increase the savings, too.	04

NIRMA UNIVERSITY
Institute of Technology
B.Tech. Computer Science and Engineering
Semester-VII
Department Elective-V

L	T	P	C
2	0	2	3

Course Code	2CSDEXX
Course Title	Ethical Hacking and Vulnerability Assessment

Course Outcomes:

At the end of the course, students will be able to -

1. summarize the core concepts related to malware, hardware and software vulnerabilities and their causes
2. choose state-of-the-art tools to exploit the vulnerabilities related to computer system and networks
3. experiment with various tools to exploit web applications
4. solve the security issues in web applications

Syllabus

Teaching Hours: 30

Unit I **05**

Working of Hackers: Invading PCs, Script Kiddies, Working of Personal Hacker Protection

Working of Spyware and Antispyware: Introduction to Spywares, Detection Escapism, Invading Privacy, Hijacking home page and search pages, working of dialers, working of keyloggers and rootkits, following spyware money trail, working of anti-spyware

Websites and privacy: Working of Cookies, Web bugs, Websites, Websites building personal profiles

Dangers of Internet Search: Working of Google, Individual Know-how

Unit II **05** **Wi-Fi security dangers and protections:** Working of Wi-Fi, Invading Wi-Fi Networks, hotspots, Evil Twin Hacks and Protections

Working of Spam: Dangers of spam, Hiding identity and identification, Working 1of Anti-spam software

Denial of Service Attacks and Protection

Virtual Private Networks, Web Blocking and Parental Controls, Personal Firewalls and Proxies SQL Injection, SQL Injection, sim spoofing, ATM card skimmers, eSIMS

Unit III **06**

Phishing Attacks: Working of Phishing, following phishing money trail, protection against phishing attacks

Zombies and Trojan Horses: Working of Zombies and Bot Networks, Working of Trojan Horses, Zombie Money Trail, Working of Zombie and Trojan Protection

Security Dangers in Browsers: Hackers exploit Networks, Protection against browser based attacks

Worms and viruses: Working of viruses and worms, antivirus software

Unit IV **04**

Vulnerability assessment: Nessus, OpenVAS, Nexpose, web application scanning tools

Penetration testing tools: Metasploit, Canvas, Writing custom exploits

Unit V **03**

Defense in Depth: Host-based and Network-based defenses (Firewalls, Intrusion Detection/Prevention)

Unit V **03**

Network analysis: TcpDump, Wireshark, Netflow

Securing and hardening systems: Bastille, CIS, MS Baseline

Unit VI **02**

Incident response and investigation: Log review, Log management and correlation, incident response process and tools

Unit VII **02**

Cloud security: Tools to assess and monitor cloud-based system security

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Preston Galla, How Personal and Internet Security Work, Que Publications
2. Alfred Basta and Wolf Halton, Computer Security Concepts, Issues and Implementation, Cengage Learning
3. Shon Harris, Allen Harper, Chris Eagle and Jonathan Ness, Gray Hat Hacking: The Ethical Hackers' Handbook, TMH Edition
4. Jon Erickson, Hacking: The Art of Exploitation, SPD
5. Peltier, T. R., Peltier, J., & Blackley, J. A., Managing a Network Vulnerability Assessment. CRC Press
6. Caswell, B., Beale, J., Ramirez, G., & Rathaus, N., Nessus, Snort, and Ethereal Power Tools: Customizing Open Source Security Applications, Elsevier

L=Lecture, T=Tutorial, P=Practical, C=Credit

^this is not an exhaustive list.

Suggested List of Experiments:

S. No.	Title	Hours
1	To implement fake authentication on an application	02
2	To crack the WPA-2 key using aircrack – ng	02
3	To perform ARP Spoofing attack	02
4	To detect ARP Poisoning attacks and other suspicious activities in the network using Wireshark	02
5	To hack a Remote Server Using a Basic Metasploit Exploit	04
6	To gain control over the target computer using client-side attacks	02
7	To perform external vulnerability scanning using Shodan, Qualys & Nmap	04
8	To perform internal vulnerability scanning using MBSA, Nmap, Nessus, Fing & Superscan & OpenVAS	04
9	To create a persistent reverse shell with Metasploit	04
10	To demonstrate security misconfiguration Attacks and Defences using a simple application / project.	04

NIRMA UNIVERSITY
Institute of Technology
B. Tech. Computer Science and Engineering
Semester-VII
Department Elective-V

L	T	P	C
2	0	2	3

Course Code	2CSDEXX
Course Title	Simulation and Mathematical Modeling

Course Outcomes:

At the end of the course, students will be able to -

1. illustrate the need of simulation and mathematical modeling in Computer Science
2. demonstrate system activities through simulation
3. apply mathematical modelling to different real life applications
4. analyze behaviour of the system under various circumstances

Syllabus	**Teaching Hours: 30**
Unit I **System Models & System Studies:** the concepts of a system, System environments, System modeling, types of models, Monte Carlo Method, Random Walks, types of system simulation, mathematical modeling cycle	**03**
Unit II **Probability Concepts in Simulation:** Stochastic variables, discrete/continuous probability functions, measures of probability, uniform random number generator, generating discrete distributions, non-uniform distributed random numbers, rejection methods	**05**
Unit III **Arrival Patterns & Service Times:** Congestion in systems, Arrival patterns, Poisson arrival patterns, Erlang/hyper-exponential/normal distribution, queuing disciplines & measures, mathematical solutions of queuing, utilization as a design factor	**04**

Unit IV **04**

Discrete System Simulation: Discrete events, generation of arrival patterns, simulation of a telephone system, delayed calls, simulation programming tasks, gathering statistics, counters & summary statistics, measuring utilization & occupancy, recording distribution & transit times, discrete simulation languages

Unit V **04**

Analysis of Simulation Output: Nature of the problem, estimation methods, simulation run statistics, replication of runs, elimination of initial bias, batch means, regenerative techniques, time series analysis, spectral analysis, autoregressive process

Unit VI **04**

Simulation Tools and Software: Study of FSM and Harel's state charts: describing a system as a state machine, developing state charts using Matlab/Simulink tool, Synchronous reactive model building using Esterel

Unit VII **06**

Optimization techniques, Applications in different domains such as biology, sports, economics etc. **Self-Study:**

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Sheldon Ross,Simulation , Academic Press.
2. Law & Kelton, Simulation Modeling & Analysis, Tata Mcgraw Hill
3. Vahid & Givargis, Embedded Systems Design-A Unified Hardware/Software Approach, Wiley Productions
4. Geoffrey Gordon, System Simulation, PHI
5. Kai Velten, Mathematical Modeling and Simulation: Introduction for Scientists and Engineers, Wiley.

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^This is not an exhaustive list.

Suggested List of Experiments:

Sr. No	Title	Hours
1	To implement a Monte Carlo simulation for estimating the value of Pi.	02

2	To implement a random number generator (random variate) following a specific distribution.	02
3	To simulate a single server queuing model.	02
4	To simulating a multi-server queuing model and calculate its performance parameters.	02
5	To implement discrete event system simulation for telephone example.	04
6	To simulate an elevator system using Harel's statecharts	04
7	To simulate Esterel based synchronous events.	04
8	To simulate an inventory management system.	04
9	To simulate an optimized resource allocation technique.	04
10	To analyse an auto regressive model for time series data.	02

NIRMA UNIVERSITY
Institute of Technology
B. Tech. Computer Science and Engineering
Semester-VII
Department Elective-V

L	T	P	C
2	0	2	3

Course Code	2CSDEXX
Course Title	Formal Methods in Software Engineering

Course Outcomes:

At the end of the course, students will be able to -

1. explain the significance of formal methods in Software Engineering
2. infer formal specification languages based on propositional logic, predicate logic, relational calculus, and finite state machines
3. apply analysis techniques for formal specification languages with help of supporting tools
4. design formal specifications for software systems

Syllabus

Teaching Hours: 30

Unit I **04**

Introduction: Software development life cycle, Role of formal methods, Lightweight formal methods, Applications of formal methods, Classification of formal methods, Explicit vs. implicit models, executable vs. non-executable, Formal verification techniques, Levels in formal methods – Formal specification, Formal development and formal verification, Theorem provers

Stages in formal methods: Specification, Development, Verification – Signoff verification, Human-directed proof, Automated proof.

Unit II **06**

Logic and Set Theory: Propositional logic, Predicate logic, Sets and relations, Lambda calculus, Assertions, Declarations, Specifications and Code, Series and Sequence.

Unit III **06**

Formal Events: Proof of program correction, Application of Hoare logic to proof of algorithm correction, Programming approach by construction, Program refinement techniques, Verification of Sequential Programs

Unit IV **06**

Formal Specification and Analysis: Declarative modelling, Difference related to model checking, Alloy commands, Functions; predicates; facts; assertions and verifications (checks), Static vs. dynamic modelling, Simulation of an operation, Check safety properties.

Formal Development: B method, Z notation, Event-B method

Unit V **08**

Verification: Verification of Sequential Programs, Static Analysis for Verifying Contracts, Symbolic Execution for Test Generation, Model-Based Test Generation, Explicit State Model Checking, Symbolic Model Checking, Model Checking Software: State bounds & Abstraction, Static Analysis of Concurrent Systems. **Self-Study:**

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on the above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. I. Van Horebeek & J. Lewi, Algebraic Specifications in Software Engineering
2. J. V. Guttag, E. Horowitz & D. R. Musser, The Design of Data Type Specifications, Current Trends in Programming Methodology (R. T. Yeh, ed.)
3. A. Diller, Z: An Introduction to Formal Methods ,Wiley
4. M. Huth and M. Ryan. Logic in Computer Science: Modeling and Reasoning about Systems. Cambridge University Press
5. Heitmeyer, C. and Mandrioli, Formal Methods for Real-Time Computing, Wiley
6. Hinchey, M.G., and Bowen, J.P., Application of Formal Methods, PH
7. Wordsworth, J.B., Software Engineering with B, Addison-Wesley
8. Daniel Jackson, Software Abstractions, MIT Press

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list

Suggested List of Experiments:

Sr. No.	Title	Hours
1	To explore tools for applying and testing predicate logic.	02
2	To explore the tools for writing Z notation and implement formal specifications.	02
3	To study and perform conceptual modeling of requirements using logic.	04

4	To study and perform algorithmic verification (model-checking) of design/models.	04
5	To study and verify functional correctness for abstract data types and refinement.	04
6	To study and verify Hoare logic assertions, refinement of a program with respect to abstract data type specification.	04
7	To explore tools for the software development process and implement a simulation.	02
8	To perform verification of sequential programs.	04
9	To study and perform white-box testing of a given application.	02
10	To study and perform grey-box testing of a given application.	02

NIRMA UNIVERSITY
Institute of Technology
B.Tech. Computer Science and Engineering
Semester-VII
Department Elective-V

L	T	P	C
2	0	2	3

Course Code	2CSDEXX
Course Name	Blockchain Technology

Course Outcomes:

At the end of the course, students will be able to

1. summarize the concept of Blockchain technology
2. develop the structure of a Blockchain network
3. evaluate security issues relating to Blockchain and cryptocurrency
4. design the applications based on Blockchain technology

Syllabus:

Teaching Hours: 30

Unit I

Introduction to Blockchain: Need, Blockchain 1.0 to 5.0, types of blockchain, Generic elements of a blockchain, digital money to distributed ledgers, design primitives, protocols, security, consensus, permissions, and privacy. **05**

Unit II

Blockchain Architecture, Design and Consensus: Basic crypto primitives: hash, signature, hash chain to Blockchain, basic consensus mechanisms, requirements for the consensus protocol for permission less environment, PoW, PoS, PoB, PoET, and scalability aspects of Blockchain consensus protocols. **06**

Unit III

Permissioned and Public Blockchains: Design goals, Consensus protocols for Permissioned Blockchains, Hyperledger Fabric, Decomposing the consensus process, Hyperledger fabric components, Smart Contracts, Chain code design, **09**

Hybrid models (PoS and PoW)

Unit IV

Blockchain cryptography: Different techniques for Blockchain cryptography, privacy and security of Blockchain, multi-sig concept **05**

Unit V

Recent trends and research issues in Blockchain: Scalability, secure cryptographic protocols on Blockchain, multiparty communication, FinTech and adoption of blockchain technology in various applications. **05**

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

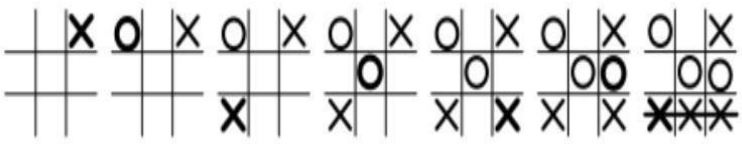
Suggested Readings^:

1. Narayanan, Arvind. et al, Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press.
2. Wattenhofer, Roger, The science of the blockchain, CreateSpace Independent Publishing Platform
3. Bahga, Arshdeep, and Vijay Madisetti,. Blockchain Applications: A Hands-on Approach, VPT
4. Nakamoto, Satoshi, Bitcoin: A peer-to-peer electronic cash system, Research Paper
5. Antonopoulos, Andreas M, Mastering Bitcoin: Programming the open blockchain, O'Reilly Media, Inc
6. Diedrich, Henning, Ethereum: Blockchains, digital assets, smart contracts, decentralized autonomous organizations, Wildfire Publishing (Sydney)
7. Draft version of "S. Shukla, M. Dhawan, S. Sharma, S. Venkatesan, 'Blockchain Technology: Cryptocurrency and Applications', Oxford University Press
8. Josh Thompson, 'Blockchain: The Blockchain for Beginnings, Guild to Blockchain Technology and Blockchain Programming', Create Space Independent Publishing Platform

L=Lecture, T=Tutorial, P=Practical, C=Credit

^this is not an exhaustive list

Suggested List of Experiments:

Sr. No	Title	Hours
1.	To implement to perform digital signature to sign and verify authenticated user. Also, show a message when tampering is detected.	02
2.	To create a blockchain and implement replay attacks on blockchain.	04
3.	To perform thorough study and installation of Anaconda 5.0.1 and Python 3.6 and perform proof of work (POW) consensus mechanism. Also, notice the changes in mining rewards and nonce requirement.	02
4.	To create a cryptocurrency and implement Byzantine Generals Problem in Python.	04
5.	To perform thorough study and installation of Remix IDE and Truffle IDE for deploying Smart Contracts and Decentralized Applications (dapps) and create and deploy a Smart Contract for any application such as finance, healthcare etc.	02
6.	To build, implement and test voting mechanism using Ethereum Blockchain. First, list the contestants on the screen and the vote they got. Whenever the user tries to vote a particular contestant, the count of the votes for the particular contestant should increase by 1. Also, the user who has already voted should be marked. Marked means "the user has already voted once and will not be allowed to vote again".	04
7.	To perform a thorough study of blockchain development on Hyperledger Fabric using Composer	02
8.	To design and develop end-to-end decentralized applications (Dapps).	04
9.	To write a Solidity contract that implements a distributed ticket sales system. Anybody can create an event (specifying the initial price and number of tickets). Anybody can then purchase one of the initial tickets or sell those tickets peer-to-peer. At the event, gate agents will check that each attendee is listed in the final attendees list on the blockchain. (Ethereum programming)	02
10.	To write a contract code to implement a two-player game (with a wager on the line) of Tic-Tac-Toe, also known as Noughts and Crosses:  (Ethereum programming)	04

NIRMA UNIVERSITY
Institute of Technology
B.Tech. Computer Science and Engineering
Semester-VII
Department Elective-V

L	T	P	C
2	0	2	3

Course Code	2CSDEXX
Course Title	Interfacing with Microprocessors

Course Outcomes:

At the end of the course, students will be able to -

1. illustrate basic architecture of microprocessors
2. utilize microcontrollers for interfacing of industrial applications
3. develop logic for programs in assembly language
4. design microprocessor-based systems for interfacing peripherals

Syllabus

Teaching Hours: 30

Unit I **08**

Microprocessor Architecture: architectural details of 8085, 8086 and 80x86 processors, pin functions, read/write machine cycles, memory organization (linear, segmentation, paging), interrupts

Unit II **09**

Microprocessor Programming: instruction set of 8086 microprocessor, programming model, modular programming (procedures and macros), BIOS/DOS interrupts and programming of interrupt service routines.

Unit III **10**

Interfacing: Architecture and interfacing of Intel 8255 Programmable Peripheral Interface, Intel 8259 Programmable Interrupt Controller, Intel 8254 programmable timer, 8237 DMA Controller,

Unit IV **03**

Analog to Digital Converter and Digital to Analog Converter., Microcontrollers, interfacing and industrial applications in process control

Self-Study:

The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents.

Laboratory Work:

Laboratory work will be based on above syllabus with minimum 10 experiments to be incorporated.

Suggested Readings^:

1. Barry B. Brey, The Intel microprocessors, Prentice Hall publisher
2. Douglas V Hall, Microprocessors and Interfacing: Programming and Hardware, McGraw Hill publisher
3. Ramesh S. Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, Prentice Hall
4. John E. Uffenbeck, Microcomputers and Microprocessors: The 8080, 8085 and Z-80 Programming, Interfacing and Troubleshooting, Prentice Hall
5. Peter Abel, IBM PC Assembly language and programming, Pearson publisher

L=Lecture, T=Tutorial, P=Practical, C=Credit

---

^this is not an exhaustive list

Suggested List of Experiments:

Sr No	Title	Hours
1.	Write an assembly program to perform addition of two numbers. Run a sample assembly program and explore different options of debug commands.	02
2.	Perform binary arithmetic operations on two 16 bit numbers and two 32 bit numbers. Write a program using 'C' to perform inline assembly	02
3.	a) Find maximum and minimum of numbers in an array. b) Add ten numbers and find their average.	04
4.	Arrange numbers in ascending order.	02
5.	a) String related interrupt handling: input, display string. b) Find the number of 1's of given 8 bit number determine its parity and display the result.	04
6.	Reverse a string entered by the user using far procedure use stack for parameter passing.	02
7.	a) Find the occurrences of a character from the entered string and display results using string related instructions. b) Write a program to find sub string from a given string using near procedure. Take the string and the substring from the user.	04
8.	Generate the Fibonacci series using far procedure and display the series.	02

9.	Check whether the number is prime, odd or even using far procedures a) in same assembly module and b) in different assembly modules c) Implement copy command for copying one file to the other. Study of 8254 timer chip and its interfacing	04
10.	Design an ISR to handle divide overflow error.	02
*11	Implement recursive procedure for finding factorial of a given number.	02