VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

# School of Computer Science and Engineering

# CURRICULUM AND SYLLABI (2023-2024)

## B. Tech. Computer Science and Engineering

## (Blockchain Technology)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

## VISION STATEMENT OF VELLORE INSTITUTE OF TECHNOLOGY

Transforming life through excellence in education and research.

## MISSION STATEMENT OF VELLORE INSTITUTE OF TECHNOLOGY

**World class Education**: Excellence in education, grounded in ethics and critical thinking, for improvement of life.

**Cutting edge Research**: An innovation ecosystem to extend knowledge and solve critical problems.

**Impactful People**: Happy, accountable, caring and effective workforce and students.

**Rewarding Co-creations**: Active collaboration with national & international industries & universities for productivity and economic development.

**Service to Society**: Service to the region and world through knowledge and compassion.

## VISION STATEMENT OF THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

To be a world-renowned centre of education, research and service in computing and allied domains.

## MISSION STATEMENT OF THE SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

- To offer computing education programs with the goal that the studentsbecome technically competent and develop lifelong learning skill.
- To undertake path-breaking research that creates new computing technologies and solutions for industry and society at large.
- To foster vibrant outreach programs for industry, research organizations, academia and society.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

# B. Tech. CSE (Blockchain Technology)

## PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. Graduates will be engineering practitioners and leaders, who would help solve industry's technological problems.
2. Graduates will be engineering professionals, innovators or entrepreneurs engaged in technology development, technology deployment, or engineering system implementation in industry.
3. Graduates will function in their profession with social awareness and responsibility.
4. Graduates will interact with their peers in other disciplines in industry and society and contribute to the economic growth of the country.
5. Graduates will be successful in pursuing higher studies in engineering or management.
6. Graduates will pursue career paths in teaching or research.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

# B. Tech. CSE (Blockchain Technology)

## PROGRAMME OUTCOMES (POs)

PO_01: Having an ability to apply mathematics and science in engineering applications.

PO_02: Having a clear understanding of the subject related concepts and of contemporary issues and apply them to identify, formulate and analyse complex engineering problems.

PO_03: Having an ability to design a component or a product applying all the relevant standards and with realistic constraints, including public health, safety, culture, society and environment

PO_04: Having an ability to design and conduct experiments, as well as to analyse and interpret data, and synthesis of information

PO_05: Having an ability to use techniques, skills, resources and modern engineering and IT tools necessary for engineering practice

PO_06: Having problem solving ability- to assess social issues (societal, health, safety, legal and cultural) and engineering problems

PO_07: Having adaptive thinking and adaptability in relation to environmental context and sustainable development

PO_08: Having a clear understanding of professional and ethical responsibility

PO_09: Having cross cultural competency exhibited by working as a member or in teams

PO_10: Having a good working knowledge of communicating in English – communication with engineering community and society

PO_11: Having a good cognitive load management skills related to project management and finance

PO_12: Having interest and recognize the need for independent and lifelong learning

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

# B. Tech. CSE (Blockchain Technology)

## PROGRAMME SPECIFIC OUTCOMES (PSOs)

1. Apply computing theory, languages and algorithms, as well as mathematical and statistical models, and the principles of optimization to appropriately formulate and use data analysis.

2. Apply the principles and techniques of database design, administration, and implementation to enhance data collection capabilities and decision-support systems. Ability to critique the role of information and analytics in supporting business processes and functions.

3. Invent and use appropriate models of data analysis, assess the quality of input, derive insight from results, and investigate potential issues. Also to organize big data sets into meaningful structures, incorporating data profiling and quality standards.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**
**B. Tech. CSE (Blockchain Technology)**

## Curriculum for 2023-2024 Batch

| Category Credit Detail | | | |
|---|---|---|---|
| **Sl.No.** | **Description** | **Credits** | **Maximum Credit** |
| 1 | FC - Foundation Core | 53 | 53 |
| 2 | DLES - Discipline-linked Engineering Sciences | 12 | 12 |
| 3 | DC - Discipline Core | 47 | 47 |
| 4 | SPE - Specialization Elective | 21 | 21 |
| 5 | PI - Projects and Internship | 9 | 9 |
| 6 | OE - Open Elective | 9 | 9 |
| 7 | BC - Bridge Course | 0 | 0 |
| 8 | NGCR - Non-graded Core Requirement | 11 | 11 |
| **Total Credits** | | 162 | |

| **Foundation Core** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **sl.no** | **Course Code** | **Course Title** | **Course Type** | **Version** | **L** | **T** | **P** | **J** | **Credits** |
| 1 | BCHY101L | Engineering Chemistry | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 2 | BCHY101P | Engineering Chemistry Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 3 | BCSE101E | Computer Programming: Python | Embedded Theory and Lab | 1.0 | 1 | 0 | 4 | 0 | 3.0 |
| 4 | BCSE102L | Structured and Object-Oriented Programming | Theory Only | 1.0 | 2 | 0 | 0 | 0 | 2.0 |
| 5 | BCSE102P | Structured and Object-Oriented Programming Lab | Lab Only | 1.0 | 0 | 0 | 4 | 0 | 2.0 |
| 6 | BCSE103E | Computer Programming: Java | Embedded Theory and Lab | 1.0 | 1 | 0 | 4 | 0 | 3.0 |
| 7 | BEEE102L | Basic Electrical and Electronics Engineering | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 8 | BEEE102P | Basic Electrical and Electronics Engineering Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 9 | BENG101L | Technical English Communication | Theory Only | 1.0 | 2 | 0 | 0 | 0 | 2.0 |
| 10 | BENG101P | Technical English Communication Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 11 | BENG102P | Technical Report Writing | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 12 | BFLE200L | B.Tech. Foreign Language - 2021onwards | Basket | 1.0 | 0 | 0 | 0 | 0 | 2.0 |
| 13 | BHSM200L | B.Tech. HSM Elective - 2021 onwards | Basket | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 14 | BMAT101L | Calculus | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 15 | BMAT101P | Calculus Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 16 | BMAT102L | Differential Equations and Transforms | Theory Only | 1.0 | 3 | 1 | 0 | 0 | 4.0 |
| 17 | BMAT201L | Complex Variables and Linear Algebra | Theory Only | 1.0 | 3 | 1 | 0 | 0 | 4.0 |
| 18 | BMAT202L | Probability and Statistics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 19 | BMAT202P | Probability and Statistics Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 20 | BPHY101L | Engineering Physics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 21 | BPHY101P | Engineering Physics Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 22 | BSTS101P | Quantitative Skills Practice I | Soft Skill | 1.0 | 0 | 0 | 3 | 0 | 1.5 |
| 23 | BSTS102P | Quantitative Skills Practice II | Soft Skill | 1.0 | 0 | 0 | 3 | 0 | 1.5 |
| 24 | BSTS201P | Qualitative Skills Practice I | Soft Skill | 1.0 | 0 | 0 | 3 | 0 | 1.5 |
| 25 | BSTS202P | Qualitative Skills Practice II | Soft Skill | 1.0 | 0 | 0 | 3 | 0 | 1.5 |

| Discipline-linked Engineering Sciences | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| sl.no | Course Code | Course Title | Course Type | Version | L | T | P | J | Credits |
| 1 | BECE102L | Digital Systems Design | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 2 | BECE102P | Digital Systems Design Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 3 | BECE204L | Microprocessors and Microcontrollers | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 4 | BECE204P | Microprocessors and Microcontrollers Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 5 | BMAT205L | Discrete Mathematics and Graph Theory | Theory Only | 1.0 | 3 | 1 | 0 | 0 | 4.0 |

| Discipline Core | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| sl.no | Course Code | Course Title | Course Type | Version | L | T | P | J | Credits |
| 1 | BCSE202L | Data Structures and Algorithms | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 2 | BCSE202P | Data Structures and Algorithms Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 3 | BCSE203E | Web Programming | Embedded Theory and Lab | 1.0 | 1 | 0 | 4 | 0 | 3.0 |
| 4 | BCSE204L | Design and Analysis of Algorithms | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 5 | BCSE204P | Design and Analysis of Algorithms Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 6 | BCSE205L | Computer Architecture and Organization | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 7 | BCSE301L | Software Engineering | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 8 | BCSE301P | Software Engineering Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 9 | BCSE302L | Database Systems | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 10 | BCSE302P | Database Systems Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 11 | BCSE303L | Operating Systems | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 12 | BCSE303P | Operating Systems Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 13 | BCSE304L | Theory of Computation | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 14 | BCSE305L | Embedded Systems | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 15 | BCSE306L | Artificial Intelligence | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 16 | BCSE307L | Compiler Design | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 17 | BCSE307P | Compiler Design Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 18 | BCSE308L | Computer Networks | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 19 | BCSE308P | Computer Networks Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 20 | BCSE309L | Cryptography and Network Security | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 21 | BCSE309P | Cryptography and Network Security Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |

| Specialization Elective | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| sl.no | Course Code | Course Title | Course Type | Version | L | T | P | J | Credits |
| 1 | BCSE324L | Foundations of Blockchain Technology | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 2 | BCSE325L | Introduction to Bitcoin | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 3 | BCSE326L | Blockchain Architecture Design | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |

| Specialization Elective | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 4 | BCSE327L | Smart Contracts | Theory Only | 1.0 | 2 | 0 | 0 | 0 | 2.0 |
| 5 | BCSE328L | Cryptocurrency Technologies | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 6 | BCSE329L | Blockchain and Distributed Ledger Technology | Theory Only | 1.0 | 2 | 0 | 0 | 0 | 2.0 |
| 7 | BCSE329P | Blockchain and Distributed Ledger Technology Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 8 | BCSE330L | Public Key Infrastructure and Trust Management | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |

| Projects and Internship | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| sl.no | Course Code | Course Title | Course Type | Version | L | T | P | J | Credits |
| 1 | BCSE399J | Summer Industrial Internship | Project | 1.0 | 0 | 0 | 0 | 0 | 1.0 |
| 2 | BCSE497J | Project - I | Project | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 3 | BCSE498J | Project - II / Internship | Project | 1.0 | 0 | 0 | 0 | 0 | 5.0 |
| 4 | BCSE499J | One Semester Internship | Project | 1.0 | 0 | 0 | 0 | 0 | 14.0 |

| Open Elective | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| sl.no | Course Code | Course Title | Course Type | Version | L | T | P | J | Credits |
| 1 | BCSE355L | AWS Solutions Architect | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 2 | BECE201L | Electronic Materials and Devices | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 3 | BECE203L | Circuit Theory | Theory Only | 1.0 | 3 | 1 | 0 | 0 | 4.0 |
| 4 | BEVD101L | Electronic Materials | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 5 | BHUM201L | Mass Communication | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 6 | BHUM202L | Rural Development | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 7 | BHUM203L | Introduction to Psychology | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 8 | BHUM204L | Industrial Psychology | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 9 | BHUM205L | Development Economics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 10 | BHUM206L | International Economics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 11 | BHUM207L | Engineering Economics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 12 | BHUM208L | Economics of Strategy | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 13 | BHUM209L | Game Theory | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 14 | BHUM210E | Econometrics | Embedded Theory and Lab | 1.0 | 2 | 0 | 2 | 0 | 3.0 |
| 15 | BHUM211L | Behavioral Economics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 16 | BHUM212L | Mathematics for Economic Analysis | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 17 | BHUM213L | Corporate Social Responsibility | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 18 | BHUM214L | Political Science | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 19 | BHUM215L | International Relations | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 20 | BHUM216L | Indian Culture and Heritage | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 21 | BHUM217L | Contemporary India | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 22 | BHUM218L | Financial Management | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 23 | BHUM219L | Principles of Accounting | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |

| Open Elective | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 24 | BHUM220L | Financial Markets and Institutions | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 25 | BHUM221L | Economics of Money, Banking and Financial Markets | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 26 | BHUM222L | Security Analysis and Portfolio Management | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 27 | BHUM223L | Options , Futures and other Derivatives | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 28 | BHUM224L | Fixed Income Securities | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 29 | BHUM225L | Personal Finance | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 30 | BHUM226L | Corporate Finance | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 31 | BHUM227L | Financial Statement Analysis | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 32 | BHUM228L | Cost and Management Accounting | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 33 | BHUM229L | Mind, Embodiment and Technology | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 34 | BHUM230L | Health Humanities in Biotechnological Era | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 35 | BHUM231L | Reproductive Choices for a Sustainable Society | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 36 | BHUM232L | Introduction to Sustainable Aging | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 37 | BHUM233L | Environmental Psychology | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 38 | BHUM234L | Indian Psychology | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 39 | BHUM235E | Psychology of Wellness | Embedded Theory and Lab | 1.0 | 2 | 0 | 2 | 0 | 3.0 |
| 40 | BHUM236L | Taxation | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 41 | BMGT108L | Entrepreneurship | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 42 | BMGT109L | Introduction to Intellectual Property | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 43 | BPHY201L | Optics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 44 | BPHY202L | Classical Mechanics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 45 | BPHY203L | Quantum Mechanics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 46 | BPHY301E | Computational Physics | Embedded Theory and Lab | 1.0 | 2 | 0 | 2 | 0 | 3.0 |
| 47 | BPHY302P | Physics Lab | Lab Only | 1.0 | 0 | 0 | 2 | 0 | 1.0 |
| 48 | BPHY401L | Solid State Physics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 49 | BPHY402L | Electromagnetic Theory | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 50 | BPHY403L | Atomic and Nuclear Physics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 51 | BPHY404L | Statistical Mechanics | Theory Only | 1.0 | 3 | 0 | 0 | 0 | 3.0 |
| 52 | BSTS301P | Advanced Competitive Coding - I | Soft Skill | 1.0 | 0 | 0 | 3 | 0 | 1.5 |
| 53 | BSTS302P | Advanced Competitive Coding - II | Soft Skill | 1.0 | 0 | 0 | 3 | 0 | 1.5 |
| 54 | CFOC105M | Emotional Intelligence | Online Course | 1.0 | 0 | 0 | 0 | 0 | 2.0 |
| 55 | CFOC133M | E-Business | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 56 | CFOC168M | Switching Circuits and Logic Design | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 57 | CFOC191M | Forests and their Management | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 58 | CFOC227M | GPU Architectures and Programming | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 59 | CFOC332M | Fundamentals of Automotive Systems | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 60 | CFOC384M | Entrepreneurship Essentials | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 61 | CFOC391M | Effective Writing | Online Course | 1.0 | 0 | 0 | 0 | 0 | 1.0 |
| 62 | CFOC469M | Financial Mathematics | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 63 | CFOC497M | Financial Statement Analysis and Reporting | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 64 | CFOC599M | Leadership and Team Effectiveness | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 65 | CFOC642M | Conservation Economics | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |

| Open Elective | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 66 | CFOC647M | Air pollution and Control | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 67 | CFOC648M | Centre-State Relations in India | Online Course | 1.0 | 0 | 0 | 0 | 0 | 2.0 |
| 68 | CFOC649M | Energy Resources, Economics, and Sustainability | Online Course | 1.0 | 0 | 0 | 0 | 0 | 2.0 |
| 69 | CFOC650M | Human Physiology | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 70 | CFOC651M | Psychology of Stress, Health and Well-being | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 71 | CFOC652M | Signal Processing Techniques and its Applications | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 72 | CFOC653M | Strength & Conditioning for the Indian Population | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 73 | CFOC654M | The Evolution of the Earth and Life | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |
| 74 | CFOC655M | United Nations Sustainable Development Goals (UN SDGs) | Online Course | 1.0 | 0 | 0 | 0 | 0 | 3.0 |

| Bridge Course | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **sl.no** | **Course Code** | **Course Title** | **Course Type** | **Version** | **L** | **T** | **P** | **J** | **Credits** |
| 1 | BENG101N | Effective English Communication | Lab Only | 1.0 | 0 | 0 | 4 | 0 | 2.0 |

| Non-graded Core Requirement | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **sl.no** | **Course Code** | **Course Title** | **Course Type** | **Version** | **L** | **T** | **P** | **J** | **Credits** |
| 1 | BCHY102N | Environmental Sciences | Online Course | 1.0 | 0 | 0 | 0 | 0 | 2.0 |
| 2 | BCSE101N | Introduction to Engineering | Project | 1.0 | 0 | 0 | 0 | 0 | 1.0 |
| 3 | BEXC100N | Extracurricular Activities / Co-Curricular Activities - B.Tech. Programmes | Basket | 1.0 | 0 | 0 | 0 | 0 | 2.0 |
| 4 | BHUM101N | Ethics and Values | Online Course | 1.0 | 0 | 0 | 0 | 0 | 2.0 |
| 5 | BSSC101N | Essence of Traditional Knowledge | Online Course | 1.0 | 0 | 0 | 0 | 0 | 2.0 |
| 6 | BSSC102N | Indian Constitution | Online Course | 1.0 | 0 | 0 | 0 | 0 | 2.0 |

| BCSE202L | Data Structures and Algorithms | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To impart basic concepts of data structures and algorithms.
2. To differentiate linear, non-linear data structures and their operations.
3. To comprehend the necessity of time complexity in algorithms.

**Course Outcomes**

On completion of this course, students should be able to:

1. Understand the fundamental analysis and time complexity for a given problem.
2. Articulate linear, non-linear data structures and legal operations permitted on them.
3. Identify and apply suitable algorithms for searching and sorting.
4. Discover various tree and graph traversals.
5. Explicate hashing, heaps and AVL trees and realize their applications.

| **Module:1** | **Algorithm Analysis** | **8 hours** |
|---|---|---|

Importance of algorithms and data structures - Fundamentals of algorithm analysis: Space and time complexity of an algorithm, Types of asymptotic notations and orders of growth - Algorithm efficiency – best case, worst case, average case - Analysis of non-recursive and recursive algorithms - Asymptotic analysis for recurrence relation: Iteration Method, Substitution Method, Master Method and Recursive Tree Method.

| **Module:2** | **Linear Data Structures** | **7 hours** |
|---|---|---|

Arrays: 1D and 2D array- Stack - Applications of stack: Expression Evaluation, Conversion of Infix to postfix and prefix expression, Tower of Hanoi – Queue - Types of Queue: Circular Queue, Double Ended Queue (deQueue) - Applications – List: Singly linked lists, Doubly linked lists, Circular linked lists- Applications: Polynomial Manipulation.

| **Module:3** | **Searching and Sorting** | **7 hours** |
|---|---|---|

Searching: Linear Search and binary search – Applications.
Sorting: Insertion sort, Selection sort, Bubble sort, Counting sort, Quick sort, Merge sort - Analysis of sorting algorithms.

| **Module:4** | **Trees** | **6 hours** |
|---|---|---|

Introduction - Binary Tree: Definition and Properties - Tree Traversals- Expression Trees:- Binary Search Trees - Operations in BST: insertion, deletion, finding min and max, finding the $k^{th}$ minimum element.

| **Module:5** | **Graphs** | **6 hours** |
|---|---|---|

Terminology – Representation of Graph – Graph Traversal: Breadth First Search (BFS), Depth First Search (DFS) - Minimum Spanning Tree: Prim's, Kruskal's - Single Source Shortest Path: Dijkstra's Algorithm.

| **Module:6** | **Hashing** | **4 hours** |
|---|---|---|

Hash functions - Separate chaining - Open hashing: Linear probing, Quadratic probing, Double hashing - Closed hashing - Random probing – Rehashing - Extendible hashing.

| **Module:7** | **Heaps and AVL Trees** | **5 hours** |
|---|---|---|

Heaps - Heap sort- Applications -Priority Queue using Heaps. AVL trees: Terminology, basic operations (rotation, insertion and deletion).

| **Module:8** | **Contemporary Issues** | **2 hours** |
|---|---|---|

| | **Total Lecture hours:** | **45 hours** |
|---|---|---|

**Text Book**

1. Mark A. Weiss, Data Structures & Algorithm Analysis in C++, $4^{th}$ Edition, 2013, Pearson Education.

| **Reference Books** | | | |
|---|---|---|---|
| 1. | Alfred V. Aho, Jeffrey D. Ullman and John E. Hopcroft, Data Structures and Algorithms, 1983, Pearson Education. | | |
| 2. | Horowitz, Sahni and S. Anderson-Freed, Fundamentals of Data Structures in C, 2008, 2nd Edition, Universities Press. | | |
| 3. | Thomas H. Cormen, C.E. Leiserson, R L. Rivest and C. Stein, Introduction to Algorithms, 2009, 3rd Edition, MIT Press. | | |
| **Mode of Evaluation**: CAT, Assignment, Quiz and FAT | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE202P | Data Structures and Algorithms Lab | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To impart basic concepts of data structures and algorithms.
2. To differentiate linear, non-linear data structures and their operations.
3. To comprehend the necessity of time complexity in algorithms.

**Course Outcomes**

On completion of this course, students should be able to:

1. Apply appropriate data structures to find solutions to practical problems.
2. Identify suitable algorithms for solving the given problems.

**Indicative Experiments**

| | |
|---|---|
| 1. | Implementation of stack data structure and its applications |
| 2. | Implementation of queue data structure and its applications |
| 3. | Implementation linked list and its application |
| 4. | Implementation of searching algorithms |
| 5. | Implementation of sorting algorithms |
| 6. | Binary Tree Traversal implementation |
| 7. | Binary Search Tree implementation |
| 8. | Graph Traversal – Depth First Search and Breadth First Search algorithm |
| 9. | Minimum Spanning Tree – Prim's and Kruskal's algorithm |
| 10. | Single Source Shortest Path Algorithm - Dijkstra's algorithm |
| | **Total Laboratory Hours** 30 hours |

**Text Book**

| | |
|---|---|
| 1. | Mark A. Weiss, Data Structures & Algorithm Analysis in C++, 2013, 4th Edition, Pearson. |

**Reference Books**

| | |
|---|---|
| 1. | Alfred V. Aho, Jeffrey D. Ullman and John E. Hopcroft, Data Structures and Algorithms, 1983, Pearson Education. |
| 2. | Horowitz, Sahni and S. Anderson-Freed, Fundamentals of Data Structures in C, 2008, 2nd Edition, Universities Press. |
| 3. | Thomas H. Cormen, C.E. Leiserson, R L. Rivest and C. Stein, Introduction to Algorithms, 2009, 3rd Edition, MIT Press. |

**Mode of assessment**: Continuous assessments and FAT.

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE204L | Design and Analysis of Algorithms | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To provide mathematical foundations for analyzing the complexity of the algorithms
2. To impart the knowledge on various design strategies that can help in solving the real world problems effectively
3. To synthesize efficient algorithms in various engineering design situations

**Course Outcomes**

On completion of this course, student should be able to:

1. Apply the mathematical tools to analyze and derive the running time of the algorithms
2. Demonstrate the major algorithm design paradigms.
3. Explain major graph algorithms, string matching and geometric algorithms along with their analysis.
4. Articulating Randomized Algorithms.
5. Explain the hardness of real-world problems with respect to algorithmic efficiency and learning to cope with it.

| **Module:1** | **Design Paradigms: Greedy, Divide and Conquer Techniques** | **6 hours** |
|---|---|---|

Overview and Importance of Algorithms - Stages of algorithm development: Describing the problem, Identifying a suitable technique, Design of an algorithm, Derive Time Complexity, Proof of Correctness of the algorithm, Illustration of Design Stages - Greedy techniques: Fractional Knapsack Problem, and Huffman coding - Divide and Conquer: Maximum Subarray, Karatsuba faster integer multiplication algorithm.

| **Module:2** | **Design Paradigms: Dynamic Programming, Backtracking and Branch & Bound Techniques** | **10 hours** |
|---|---|---|

Dynamic programming: Assembly Line Scheduling, Matrix Chain Multiplication, Longest Common Subsequence, 0-1 Knapsack, TSP- Backtracking: N-Queens problem, Subset Sum, Graph Coloring- Branch & Bound: LIFO-BB and FIFO BB methods: Job Selection problem, 0-1 Knapsack Problem

| **Module:3** | **String Matching Algorithms** | **5 hours** |
|---|---|---|

Naïve String-matching Algorithms, KMP algorithm, Rabin-Karp Algorithm, Suffix Trees.

| **Module:4** | **Graph Algorithms** | **6 hours** |
|---|---|---|

All pair shortest path: Bellman Ford Algorithm, Floyd-Warshall Algorithm - Network Flows: Flow Networks, Maximum Flows: Ford-Fulkerson, Edmond-Karp, Push Re-label Algorithm – Application of Max Flow to maximum matching problem

| **Module:5** | **Geometric Algorithms** | **4 hours** |
|---|---|---|

Line Segments: Properties, Intersection, sweeping lines - Convex Hull finding algorithms: Graham's Scan, Jarvis' March Algorithm.

| **Module:6** | **Randomized algorithms** | **5 hours** |
|---|---|---|

Randomized quick sort - The hiring problem - Finding the global Minimum Cut.

| **Module:7** | **Classes of Complexity and Approximation Algorithms** | **7 hours** |
|---|---|---|

The Class P - The Class NP - Reducibility and NP-completeness – SAT (Problem Definition and statement), 3SAT, Independent Set, Clique, Approximation Algorithm – Vertex Cover, Set Cover and Travelling salesman

| **Module:8** | **Contemporary Issues** | **2 hours** |
|---|---|---|
| | **Total Lecture hours:** | **45 hours** |

**Text Book**

1. Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, Introduction to Algorithms, Third edition, MIT Press, 2009.

| **Reference Books** | | | | |
|---|---|---|---|---|
| 1. | Jon Kleinberg and ÉvaTardos, Algorithm Design, Pearson Education, 1st Edition, 2014. | | | |
| 2. | Rajeev Motwani, Prabhakar Raghavan; Randomized Algorithms, Cambridge University Press, 1995 (Online Print – 2013) | | | |
| 3. | Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, Network Flows: Theory, Algorithms, and Applications, 1st Edition, Pearson Education, 2014. | | | |
| **Mode of Evaluation**: CAT, Written assignments, Quiz, FAT. | | | | |
| Recommended by Board of Studies | | 04-03-2022 | | |
| Approved by Academic Council | | No. 65 | Date | 17-03-2022 |

| BCSE204P | Design and Analysis of Algorithms Lab | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | **Nil** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To provide mathematical foundations for analyzing the complexity of the algorithms
2. To impart the knowledge on various design strategies that can help in solving the real world problems effectively
3. Synthesize efficient algorithms in various engineering design situations

**Course Outcome**

On completion of this course, student should be able to:
1. Demonstrate the major algorithm design paradigms.
2. Explain major graph algorithms, string matching and geometric algorithms along with their analysis.

**Indicative Experiments**

| | |
|---|---|
| 1. | Greedy Strategy : Activity Selection & Huffman coding |
| 2. | Dynamic Programming : ALS, Matrix Chain Multiplication , Longest Common Subsequence, 0-1 Knapsack |
| 3. | Divide and Conquer : Maximum Subarray and Karatsuba faster integer multiplication algorithm |
| 4. | Backtracking: N-queens |
| 5. | Branch and Bound: Job selection |
| 6 | String matching algorithms : Naïve, KMP and Rabin Karp,suffix trees |
| 7 | MST and all pair shortest path algorithms |
| 8 | Network Flows : Ford –Fulkerson and Edmond - Karp |
| 9 | Intersection of line segments &Finding Convexhull, Finding closest pair of points |
| 10 | Polynomial time algorithm for verification of NPC problems |
| 11 | Approximation and Randomized algorithms |
| | Total Laboratory Hours \| 30 Hours |

**Text Book**

| | |
|---|---|
| 1. | Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, Introduction to Algorithms, Third edition, MIT Press, 2009. |

**Reference Books**

| | |
|---|---|
| 1. | Jon Kleinberg and ÉvaTardos, Algorithm Design, Pearson Education, 1st Edition, 2014. |
| 2. | Rajeev Motwani, Prabhakar Raghavan; Randomized Algorithms, Cambridge University Press, 1995 (Online Print – 2013) |
| 3. | Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, Network Flows: Theory, Algorithms, and Applications, 1st Edition, Pearson Education, 2014. |

**Mode of assessment**: Continuous assessments, FAT.

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE205L | Computer Architecture and Organization | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus Version** | | | |
| | | **1.0** | | | |

**Course Objectives**

1. To acquaint students with the basic concepts of fundamental component, architecture, register organization and performance metrics of a computer and to impart the knowledge of data representation in binary and to understand the implementation of arithmetic algorithms in a typical computer.
2. To teach students how to describe machine capabilities and design an effective data path design for instruction execution. To introduce students to syntax and semantics of machine level programming.
3. To make students understand the importance of memory systems, IO interfacing techniques and external storage and their performance metrics for a typical computer. And explore various alternate techniques for improving the performance of a processor.

**Course Outcomes**

On completion of this course, student should be able to:

1. Differentiate Von Neumann, Harvard, and CISC and RISC architectures. Analyze the performance of machine with different capabilities. Recognize different instruction formats and addressing modes. Validate efficient algorithm for fixed point and floating point arithmetic operations.
2. Explain the importance of hierarchical memory organization. Able to construct larger memories. Analyze and suggest efficient cache mapping technique and replacement algorithms for given design requirements. Demonstrate hamming code for error detection and correction.
3. Understand the need for an interface. Compare and contrast memory mapping and IO mapping techniques. Describe and Differentiate different modes of data transfer. Appraise the synchronous and asynchronous bus for performance and arbitration.
4. Assess the performance of IO and external storage systems. Classify parallel machine models. Analyze the pipeline hazards and solutions.

| **Module:1** | **Introduction To Computer Architecture and Organization** | **5 Hours** |
|---|---|---|

Overview of Organization and Architecture –Functional components of a computer: Registers and register files - Interconnection of components - Overview of IAS computer function - Organization of the von Neumann machine - Harvard architecture - CISC & RISC Architectures.

| **Module:2** | **Data Representation and Computer Arithmetic** | **5 Hours** |
|---|---|---|

Algorithms for fixed point arithmetic operations: Multiplication (Booths, Modified Booths), Division (restoring and non-restoring) - Algorithms for floating point arithmetic operations - Representation of nonnumeric data (character codes).

| **Module:3** | **Instruction Sets and Control Unit** | **9 Hours** |
|---|---|---|

Computer Instructions: Instruction sets, Instruction Set Architecture, Instruction formats, Instruction set categories - Addressing modes - Phases of instruction cycle – ALU - Data-path and control unit: Hardwired control unit and Micro programmed control unit - Performance metrics: Execution time calculation, MIPS, MFLOPS.

| **Module:4** | **Memory System Organization and Architecture** | **7 Hours** |
|---|---|---|

Memory systems hierarchy: Characteristics, Byte Storage methods, Conceptual view of memory cell - Design of scalable memory using RAM's- ROM's chips - Construction of larger size memories - Memory Interleaving - Memory interface address map- Cache memory: principles, Cache memory management techniques, Types of caches, caches misses, Mean

| | | |
|---|---|---|
| memory access time evaluation of cache. | | |
| | | |
| **Module:5** | **Interfacing and Communication** | **5 Hours** |
| I/O fundamentals: handshaking, buffering, I/O Modules - I/O techniques: Programmed I/O, Interrupt-driven I/O, Direct Memory Access, Direct Cache Access - Interrupt structures: Vectored and Prioritized-interrupt overhead - Buses: Synchronous and asynchronous - Arbitration. | | |
| | | |
| **Module:6** | **Subsystems** | **5 Hours** |
| External storage systems: Solid state drivers - Organization and Structure of disk drives: Electronic- magnetic and optical technologies - Reliability of memory systems - Error detecting and error correcting systems - RAID Levels - I/O Performance | | |
| | | |
| **Module:7** | **High Performance Processors** | **7 Hours** |
| Classification of models - Flynn's taxonomy of parallel machine models (SISD, SIMD, MISD, MIMD) - Pipelining: Two stages, Multi stage pipelining, Basic performance issues in pipelining, Hazards, Methods to prevent and resolve hazards and their drawbacks - Approaches to deal branches - Superscalar architecture: Limitations of scalar pipelines, superscalar versus super pipeline architecture, superscalar techniques, performance evaluation of superscalar architecture - performance evaluation of parallel processors: Amdahl's law, speed-up and efficiency. | | |
| | | |
| **Module:8** | **Contemporary Issues** | **2 Hours** |
| | | |
| | **Total Lecture Hours** | **45 Hours** |
| **Text Book(s)** | | |
| 1 | David A. Patterson and John L. Hennessy, Computer Organization and Design -The Hardware / Software Interface 6th Edition, Morgan Kaufmann, 2020 | |
| **Reference Book(s)** | | |
| 1 | Computer Architecture and Organization-Designing for Performance, William Stallings, Tenth edition, Pearson Education series, 2016 | |
| 2 | Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer organization, Mc Graw Hill, Fifth edition, Reprint 2011. | |
| **Mode of Evaluation:** CAT, Written Assignments, Quiz and FAT. | | |

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE301L | Software Engineering | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To introduce the essential Software Engineering concepts.
2. To impart concepts and skills for performing analysis, design ,develop, test and evolve efficient software systems of various disciplines and applications
3. To make familiar about engineering practices, standards and metrics for developing software components and products.

**Course Outcomes**

On completion of this course, student should be able to:

1. Apply and assess the principles of various process models for the software development.
2. Demonstrate various software project management activities that include planning , Estimations, Risk assessment and Configuration Management
3. Perform Requirements modelling and apply appropriate design and testing heuristics to produce quality software systems.
4. Demonstrate the complete Software life cycle activities from requirements analysis to maintenance using the modern tools and techniques.
5. Escalate the use of various standards and metrics in evaluating the process and product.

| **Module:1** | **Overview Of Software Engineering** | **6 hours** |
|---|---|---|

Nature of Software, Software Engineering, Software process, project, product, Process Models

Classical Evolutionary models, Introduction to Agility - Agile Process-Extreme programming - XP Process – Principles of Agile Software Development framework - Overview of System Engineering

| **Module:2** | **Introduction To Software Project Management** | **6 hours** |
|---|---|---|

Planning, Scope, Work break-down structure, Milestones, Deliverables, Cost and Estimates - (Human Resources, Time-scale, Costs), Risk Management, RMMM Plan, CASE TOOLS, Agile Project Management, Managing team dynamics and communication, Metrics and Measurement

| **Module:3** | **Modelling Requirements** | **8 hours** |
|---|---|---|

Software requirements and its types, Requirements Engineering process, Requirement Elicitation, System Modeling – Requirements Specification and Requirement Validation, Requirements Elicitation techniques, Requirements management in Agile.

| **Module:4** | **Software Design** | **8 hours** |
|---|---|---|

Design concepts and principles - Abstraction - Refinement - Modularity Cohesion coupling, Architectural design, Detailed Design Transaction Transformation, Refactoring of designs, Object oriented Design User-Interface Design

| **Module:5** | **Validation And Verification** | **7 hours** |
|---|---|---|

Strategic Approach to Software Testing, Testing Fundamentals Test Plan, Test Design, Test Execution, Reviews, Inspection and Auditing – Regression Testing – Mutation Testing - Object oriented testing - Testing Web based System - Mobile App testing – Mobile test Automation and tools – DevOps Testing – Cloud and Big Data Testing

| **Module:6** | **Software Evolution** | **4 hours** |
|---|---|---|

| | | |
|---|---|---|
| Software Maintenance, Types of Maintenance, - Software Configuration Management – Overview – SCM Tools. Re-Engineering, Reverse Engineering, Software Reuse | | |
| **Module:7** | **Quality Assurance** | **4 hours** |
| Product and Process Metrics, Quality Standards Models ISO, TQM, Six-Sigma, Process improvement Models: CMM & CMMI. Quality Control and Quality Assurance - Quality Management - Quality Factors - Methods of Quality Management | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | **Total Lecture hours:** | **45 hours** |

| | |
|---|---|
| **Text Book(s)** | |
| 1. | Ian Somerville, Software Engineering, 10th Edition, Addison-Wesley, 2015 |
| **Reference Books** | |
| 1. | Roger S. Pressman and Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 10th edition, McGraw Hill Education, 2019 |
| 2. | William E. Lewis , Software Testing and Continuous Quality Improvement, Third Edition, Auerbach Publications, 2017 |

| | | | |
|---|---|---|---|
| Mode of Evaluation: CAT, Written assignment, Quiz, FAT. | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE301P | Software Engineering Lab | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To introduce the essential Software Engineering concepts.
2. To impart concepts and skills for performing analysis, design ,develop, test and evolve efficient software systems of various disciplines and applications
3. To make familiar about engineering practices, standards and metrics for developing software components and products.

**Course Outcome**

On completion of this course, student should be able to:

1. Demonstrate the complete Software life cycle activities from requirements analysis to maintenance using the modern tools and techniques.

| **Indicative Experiments** | |
|---|---|
| 1. | Analysis and Identification of the suitable process models |
| 2. | Work Break-down Structure (Process Based, Product Based, Geographic Based and Role Based) and Estimations |
| 3. | Requirement modelling using Entity Relationship Diagram(Structural Modeling) |
| 4. | Requirement modelling using Context flow diagram, DFD ( Functional Modeling) |
| 5. | Requirement modelling using State Transition Diagram ( Behavioral Modeling) |
| 6. | OO design – Use case Model, Class Model |
| 7. | OO design – Interaction Models |
| 8. | OO design – Package, Component and deployment models |
| 9. | Design and demonstration of test cases. Functional Testing and Non- Functional Testing (using any open source tools) |
| 10. | Story Boarding and User Interface design Modelling |
| | Total Laboratory Hours: 30 hours |

**Text Book(s)**

1. Ian Somerville, Software Engineering, 10th Edition, Addison-Wesley, 2015

**Reference Books**

1. Roger S. Pressman and Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 10th edition, McGraw Hill Education, 2019
2. William E. Lewis, Software Testing and Continuous Quality Improvement, Third Edition, Auerbach Publications, 2017

**Mode of assessment:** Continuous assessments, FAT.

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE302L | Database Systems | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To understand the concepts of File system and structure of the database; Designing an Entity-Relationship model for a real-life application and Mapping a database schema from the ER model.
2. To differentiate various normal forms, evaluate relational schemas for design qualities and optimize a query.
3. To impart the working methodologies of transaction management, understand concurrency control, recovery, indexing, access methods and fundamental view on unstructured data and its management.

**Course Outcomes**

On completion of this course, student should be able to:

1. Comprehend the role of database management system in an organization and design the structure and operation of the relational data model.
2. Develop a database project depending on the business requirements, considering various design issues.
3. List the concepts of indexing and accessing methods.
4. Explain the concept of a database transaction processing and comprehend the concept of database facilities including concurrency control, backup and recovery.
5. Review the fundamental view on unstructured data and describe other emerging database technologies.

| **Module:1** | **Database Systems Concepts and Architecture** | **4 hours** |
|---|---|---|

Need for database systems – Characteristics of Database Approach – Advantages of using DBMS approach - Actors on the Database Management Scene: Database Administrator - Classification of database management systems - Data Models - Schemas and Instances - Three-Schema Architecture - The Database System Environment - Centralized and Client/Server Architectures for DBMSs – Overall Architecture of Database Management Systems

| **Module:2** | **Relational Model and E-R Modeling** | **6 hours** |
|---|---|---|

Relational Model: Candidate Keys, Primary Keys, Foreign Keys - Integrity Constraints - Handling of Nulls - Entity Relationship Model: Types of Attributes, Relationships, Structural Constraints, Relational model Constraints – Mapping ER model to a relational schema – Extended ER Model - Generalization – Specialization – Aggregations.

| **Module:3** | **Relational Database Design** | **6 hours** |
|---|---|---|

Database Design – Schema Refinement - Guidelines for Relational Schema - Functional dependencies - Axioms on Functional Dependencies- Normalization: First, Second and Third Normal Forms - Boyce Codd Normal Form, Multi-valued dependency and Fourth Normal form - Join dependency and Fifth Normal form

| **Module:4** | **Physical Database Design and Query Processing** | **8 hours** |
|---|---|---|

File Organization - Indexing: Single level indexing, multi-level indexing, dynamic multilevel Indexing - B+ Tree Indexing – Hashing Techniques: Static and Dynamic Hashing – Relational Algebra - Translating SQL Queries into Relational Algebra - Query Processing – Query Optimization: Algebraic Query Optimization, Heuristic query optimization Rules, Join Query Optimization using Indexing and Hashing - Tuple Relational Calculus.

| **Module:5** | **Transaction Processing and Recovery** | **8 hours** |
|---|---|---|

| | | |
|---|---|---|
| Introduction to Transaction Processing – Transaction concepts: ACID Properties of Transactions, Transaction States - Serial and Serializable Schedules - Schedules based on recoverability – Schedules based on Serializability - Conflict Serializabilty - Recovery Concepts: Log Based Recovery Protocols, Recovery based on deferred update, Recovery techniques based on immediate update – Shadow Paging Algorithm | | |
| **Module:6** | **Concurrency Control In Transaction Processing** | **8 hours** |
| Concurrent Transactions – Lost Update Problem - Concurrency Control Techniques: Time Stamp Based Protocols, Thomas Write Rule, Lock Based Protocols, Lock Compatibility Matrix, - Two-Phase Locking Protocol - Lock Conversions - Graph Based Protocols for Concurrency Control - Tree Protocol for Concurrency Control – Deadlocks Based on Locks in Transactions – Deadlock Handling Techniques – Transaction Deadlock Detection Techniques – Transaction Deadlock Prevention Techniques – Multi-Granularity Locking for avoiding Transaction Deadlocks | | |
| **Module:7** | **NOSQL Database Management** | **3 hours** |
| Introduction, Need of NoSQL, CAP Theorem, different NoSQL data bases: Key-value data stores, Columnar families, Document databases, Graph databases | | |
| **Module:8** | **Contemporary Issues** | **2 Hours** |
| | | |
| | **Total Lecture hours:** | **45 hours** |

| | |
|---|---|
| **Text Book** | |
| 1. | R. Elmasri & S. B. Navathe, Fundamentals of Database Systems, Addison Wesley, 7th Edition, 2016 |
| | |
| **Reference Books** | |
| 1. | A. Silberschatz, H. F. Korth & S. Sudarshan, Database System Concepts, McGraw Hill, 7th Edition 2019. |
| 2. | Raghu Ramakrishnan, Database Management Systems, Mcgraw-Hill, 4th Edition, 2018 |
| 3. | C.J.Date, A.Kannan, S.Swamynathan," An Introduction to Database Systems", Pearson, Eighth Edition, 2006. |
| 4. | Gerardus Blokdyk, NoSQL Databases A Complete Guide, 5STARCooks, 2021 |
| **Mode of Evaluation**: CAT, Written assignments, Quiz and FAT. | |

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE302P | Database Systems Lab | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. Basic ability to understand the concepts of File system and structure of the database; Designing an Entity-Relationship model for a real-life application and Mapping a database schema from the ER model.
2. Differentiate various normal forms, evaluate relational schemas for design qualities and optimize a query.
3. Explain the working methodologies of transaction management and give a solution during a transaction failure. Understand the basic concepts on concurrency control, recovery, indexing, access methods and fundamental view on unstructured data and its management.

**Course Outcome**

On completion of this course, student should be able to:

1. Design the structure and operation of the relational data model.
2. Examine the data requirements of the real world and design a database management system.

| **Indicative Experiments** | |
|---|---|
| 1. | Data Definition and Data Manipulation Language |
| 2. | Constraints |
| 3. | Single row functions |
| 4. | Operators and group functions |
| 5. | Sub query, views and joins |
| 6. | High Level Language Extensions - Procedures, Functions, Cursors and Triggers |
| | **Total Laboratory Hours** 30 hours |

**Text Book**

1. R. Elmasri & S. B. Navathe, Fundamentals of Database Systems, Addison Wesley, 7th Edition, 2016

**Reference Books**

1. A. Silberschatz, H. F. Korth & S. Sudarshan, Database System Concepts, McGraw Hill, 7th Edition 2019.
2. Raghu Ramakrishnan, Database Management Systems, Mcgraw-Hill, 4th Edition, 2018
3. C.J.Date, A.Kannan, S.Swamynathan," An Introduction to Database Systems", Pearson, Eighth Edition, 2006.
4. Gerardus Blokdyk, NoSQL Databases A Complete Guide, 5STARCooks, 2021

**Mode of assessment**: Continuous assessments, FAT

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE303L | Operating Systems | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To introduce the operating system concepts, designs and provide skills required to implement the services.
2. To describe the trade-offs between conflicting objectives in large scale system design.
3. To develop the knowledge for application of the various design issues and services.

**Course Outcomes**

On completion of this course, student should be able to:

1. Interpret the evolution of OS functionality, structures, layers and apply various types of system calls of various process states.
2. Design scheduling algorithms to compute and compare various scheduling criteria.
3. Apply and analyze communication between inter process and synchronization techniques.
4. Implement page replacement algorithms, memory management problems and segmentation.
5. Differentiate the file systems for applying different allocation, access technique, representing virtualization and providing protection and security to OS.

| **Module:1** | **Introduction** | **3 hours** |
|---|---|---|

Introduction to OS: Functionality of OS - OS design issues - Structuring methods (monolithic, layered, modular, micro-kernel models) - Abstractions, processes, resources - Influence of security, networking, and multimedia.

| **Module:2** | **OS Principles** | **4 hours** |
|---|---|---|

System calls, System/Application Call Interface – Protection: User/Kernel modes - Interrupts -Processes - Structures (Process Control Block, Ready List etc.), Process creation, management in Unix – Threads: User level, kernel level threads and thread models.

| **Module:3** | **Scheduling** | **9 hours** |
|---|---|---|

Processes Scheduling - CPU Scheduling: Pre-emptive, non-pre-emptive - Multiprocessor scheduling – Deadlocks - Resource allocation and management - Deadlock handling mechanisms: prevention, avoidance, detection, recovery.

| **Module:4** | **Concurrency** | **8 hours** |
|---|---|---|

Inter-process communication, Synchronization - Implementing synchronization primitives (Peterson's solution, Bakery algorithm, synchronization hardware) - Semaphores – Classical synchronization problems, Monitors: Solution to Dining Philosophers problem – IPC in Unix, Multiprocessors and Locking - Scalable Locks - Lock-free coordination.

| **Module:5** | **Memory Management** | **7 hours** |
|---|---|---|

Main memory management, Memory allocation strategies, Virtual memory: Hardware support for virtual memory (caching, TLB) – Paging - Segmentation - Demand Paging - Page Faults - Page Replacement -Thrashing - Working Set.

| **Module:6** | **Virtualization and File System Management** | **6 hours** |
|---|---|---|

Virtual Machines - Virtualization (Hardware/Software, Server, Service, Network - Hypervisors - Container virtualization - Cost of virtualization - File system interface (access methods, directory structures) - File system implementation (directory implementation, file allocation methods) - File system recovery - Journaling - Soft updates - Log-structured file system - Distributed file system.

| **Module:7** | **Storage Management, Protection and Security** | **6 hours** |
|---|---|---|

Disk structure and attachment – Disk scheduling algorithms (seek time, rotational latency based)- System threats and security – Policy vs mechanism - Access vs authentication -

| | | |
|---|---|---|
| System protection: Access matrix – Capability based systems - OS: performance, scaling, future directions in mobile OS. | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | | |
| | | |
| | **Total Lecture hours:** | **45 hours** |
| **Text Book** | | |
| 1. | Abraham Silberschatz, Peter B. Galvin, Greg Gagne, "Operating System Concepts", 2018, 10th Edition, Wiley, United States. | |
| **Reference Books** | | |
| 1. | Andrew S. Tanenbaum, "Modern Operating Systems", 2016, 4th Edition, Pearson, United Kingdom. | |
| 2. | William Stallings, "Operating Systems: Internals and Design Principles", 2018, 9th Edition, Pearson, United Kingdom. | |
| **Mode of Evaluation**: CAT, Written Assignment, Quiz, FAT | | |
| Recommended by Board of Studies | 04-03-2022 | |
| Approved by Academic Council | No. 65 | Date 17-03-2022 |

| BCSE303P | Operating Systems Lab | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | **Nil** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To introduce the operating system concepts, designs and provide skills required to implement the services.
2. To describe the trade-offs between conflicting objectives in large scale system design.
3. To develop the knowledge for application of the various design issues and services.

**Course Outcome**

On completion of this course, student should be able to:

1. Interpret the evolution of OS functionality, structures, layers and apply various types of system calls of various process states.
2. Design scheduling algorithms to compute and compare various scheduling criteria.
3. Apply and analyze communication between inter process and synchronization techniques.
4. Implement page replacement algorithms, memory management problems and segmentation.
   Differentiate the file systems for applying different allocation, access technique, representing virtualization and providing protection and security to OS.

**Indicative Experiments**

| | |
|---|---|
| 1. | Study of Basic Linux Commands |
| 2. | Implement your own bootloader program that helps a computer to boot an OS. |
| 3. | Shell Programming (I/O, Decision making, Looping, Multi-level branching) |
| 4. | Creating child process using fork () system call, Orphan and Zombie process creation |
| 5. | Simulation of CPU scheduling algorithms (FCFS, SJF, Priority and Round Robin) |
| 6. | Implement process synchronization using semaphores / monitors. |
| 7. | Simulation of Banker s algorithm to check whether the given system is in safe state or not. Also check whether addition resource requested can be granted immediately |
| 8. | Parallel Thread management using Pthreads library. Implement a data parallelism using multi-threading |
| 9. | Dynamic memory allocation algorithms - First-fit, Best-fit, Worst-fit algorithms |
| 10. | Page Replacement Algorithms FIFO, LRU and Optimal |
| 11. | Implement a file locking mechanism. |
| 12. | Virtualization Setup: Type-1, Type-2 Hypervisor (Detailed Study Report) |
| | **Total Laboratory Hours** 30 hours |

**Text Book**

| | |
|---|---|
| 1. | Fox, Richard, "Linux with Operating System Concepts", 2022, 2nd Edition, Chapman and Hall/CRC, UK. |

**Reference Books**

| | |
|---|---|
| 1. | Love, Robert, "Linux System Programming: talking directly to the kernel and C library", 2013, 2nd Edition, O'Reilly Media, Inc, United States. |
| 2. | Abraham Silberschatz, Peter B. Galvin, Greg Gagne, "Operating System Concepts", 2018, 10th Edition, Wiley, United States. |

**Mode of Assessment**: Continuous Assessments, FAT

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE304L | Theory of Computation | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **Nil** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. Types of grammars and models of automata.
2. Limitation of computation: What can be and what cannot be computed.
3. Establishing connections among grammars, automata and formal languages.

**Course Outcome**

On completion of this course, student should be able to:
1. Compare and analyse different computational models
2. Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.
3. Identify limitations of some computational models and possible methods of proving them.
4. Represent the abstract concepts mathematically with notations.

| | | |
|---|---|---|
| **Module:1** | **Introduction to Languages and Grammars** | **4 hours** |
| Recall on Proof techniques in Mathematics - Overview of a Computational Models - Languages and Grammars - Alphabets - Strings - Operations on Languages, Overview on Automata | | |
| **Module:2** | **Finite State Automata** | **8 hours** |
| Finite Automata (FA) - Deterministic Finite Automata (DFA) - Non-deterministic Finite Automata (NFA) - NFA with epsilon transitions – NFA without epsilon transition, conversion of NFA to DFA, Equivalence of NFA and DFA – minimization of DFA | | |
| **Module:3** | **Regular Expressions and Languages** | **7 hours** |
| Regular Expression - FA and Regular Expressions: FA to regular expression and regular expression to FA - Pattern matching and regular expressions - Regular grammar and FA - Pumping lemma for regular languages - Closure properties of regular languages | | |
| **Module:4** | **Context Free Grammars** | **7 hours** |
| Context-Free Grammar (CFG) – Derivations - Parse Trees - Ambiguity in CFG - CYK algorithm – Simplification of CFG – Elimination of Useless symbols, Unit productions, Null productions - Normal forms for CFG: CNF and GNF - Pumping Lemma for CFL - Closure Properties of CFL | | |
| **Module:5** | **Pushdown Automata** | **5 hours** |
| Definition of the Pushdown automata - Languages of a Pushdown automata – Power of Non-Deterministic Pushdown Automata and Deterministic pushdown automata | | |
| **Module:6** | **Turing Machine** | **6 hours** |
| Turing Machines as acceptor and transducer - Multi head and Multi tape Turing Machines – Universal Turing Machine - The Halting problem - Turing-Church thesis | | |
| **Module:7** | **Recursive and Recursively Enumerable Languages** | **6 hours** |
| Recursive and Recursively Enumerable Languages, Language that is not Recursively Enumerable (RE) – computable functions – Chomsky Hierarchy – Undecidable problems - Post's Correspondence Problem | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | | |
| | **Total Lecture hours:** | **45 hours** |

**Text Book**

1. J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computation", Third Edition, Pearson Education, India 2008. ISBN: 978-8131720479

**Reference Books**

| | | | |
|---|---|---|---|
| 1. | Peter Linz, "An Introduction to Formal Languages and Automata", Sixth Edition, Jones & Bartlett, 2016. ISBN: 978-9384323219 | | |
| 2. | K. Krithivasan and R. Rama, "Introduction to Formal Languages, Automata and Computation", Pearson Education, 2009. ISBN: 978-8131723562 | | |
| **Mode of Evaluation**: CAT, Assignment, Quiz, FAT. | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE305L | Embedded Systems | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To expose students to various challenges and constraints of special purpose computing systems in terms of resources and functional requirements.
2. To introduce students to various components of typical embedded systems viz., sensors and actuators, data converters, UART etc., their interfacing, programming environment for developing any smart systems and various serial communication protocols for optimal components interfacing and communication.
3. To make students understand the importance of program modeling, optimization techniques and debugging tools for product development and explore various solutions for real time scheduling issues in terms of resources and deadline.

**Course Outcomes**

On completion of this course, students should be able to:

1. Identify the challenges in designing an embedded system using various microcontrollers and interfaces.
2. To summaries the functionality of any special purpose computing system, and to propose smart solutions to engineering challenges at the prototype level.
3. To examine the working principle and interface of typical embedded system components, create programme models, apply various optimization approaches including simulation environment and demonstration using debugging tools.
4. To evaluate the working principle of serial communication protocols and their proper use, as well as to analyze the benefits and drawbacks of real-time scheduling algorithms and to recommend acceptable solutions for specific challenges.

| | | |
|---|---|---|
| **Module:1** | **Introduction** | **5 hours** |
| Overview of Embedded Systems, Design challenges, Embedded processor technology, Hardware Design, Micro-controller architecture -8051, PIC, and ARM. | | |
| **Module:2** | **I/O Interfacing Techniques** | **8 hours** |
| Memory interfacing, A/D, D/A, Timers, Watch-dog timer, Counters, Encoder & Decoder, UART, Sensors and actuators interfacing. | | |
| **Module:3** | **Architecture of Special Purpose Computing System** | **6 hours** |
| ATM, Handheld devices, Data Compressor, Image Capturing Devices–Architecture and Requirements, Challenges & Constraints of special purpose computing system. | | |
| **Module:4** | **Programming Tools** | **7 hours** |
| Evolution of embedded programming tools, Modelling programs, Code optimization, Logic analyzers, Programming environment. | | |
| **Module:5** | **Real Time Operating System** | **8 hours** |
| Classification of Real time system, Issues & challenges in RTS, Real time scheduling schemes- EDF-RMS & Hybrid techniques, eCOS, POSIX, Protothreads. | | |
| **Module:6** | **Embedded Networking Protocols** | **5 hours** |
| Inter Integrated Circuits (I2C), Controller Area Network, Embedded Ethernet Controller, RS232, Bluetooth, Zigbee, Wifi. | | |
| **Module:7** | **Applications of Embedded Systems** | **4 hours** |
| Introduction to embedded system applications using case studies – Role in Agriculture sector, Automotive electronics, Consumer Electronics, Industrial controls, Medical Electronics. | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |

| | | |
|---|---|---|
| | **Total Lecture hours:** | **45 hours** |
| **Text Book** | | |
| 1. | Marilyn Wolf, Computers as Components – Principles of Embedded Computing System Design, Fourth Edition, Morgan Kaufman Publishers, 2016. | |
| **Reference Books** | | |
| 1. | Embedded Systems Architecture, Programming and Design, by Raj Kamal, McGraw Hill Education, 3e, 2015. | |
| 2. | Embedded System Design A Unified Hardware/Sofware Introduction, by Vahid G Frank and Givargis Tony, John Wiley & Sons, 2009. | |
| **Mode of Evaluation**: CAT, written assignment, Quiz, FAT. | | |
| Recommended by Board of Studies | 04-03-2022 | |
| Approved by Academic Council | No. 65 | Date 17-03-2022 |

| BCSE306L | Artificial Intelligence | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To impart artificial intelligence principles, techniques and its history.
2. To assess the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, and learning methods in solving engineering problems
3. To develop intelligent systems by assembling solutions to concrete computational problems

**Course Outcomes**

On completion of this course, student should be able to:

1. Evaluate Artificial Intelligence (AI) methods and describe their foundations.
2. Apply basic principles of AI in solutions that require problem-solving, inference, perception, knowledge representation and learning.
3. Demonstrate knowledge of reasoning, uncertainty, and knowledge representation for solving real-world problems
4. Analyse and illustrate how search algorithms play a vital role in problem-solving

| | | |
|---|---|---|
| **Module:1** | **Introduction** | **6 hours** |
| Introduction- Evolution of AI, State of Art -Different Types of Artificial Intelligence- Applications of AI-Subfields of AI-Intelligent Agents- Structure of Intelligent Agents- Environments | | |
| **Module:2** | **Problem Solving based on Searching** | **6 hours** |
| Introduction to Problem Solving by searching Methods-State Space search, Uninformed Search Methods – Uniform Cost Search, Breadth First Search- Depth First Search-Depth-limited search, Iterative deepening depth-first, Informed Search Methods- Best First Search, A* Search | | |
| **Module 3** | **Local Search and Adversarial Search** | **5 hours** |
| Local Search algorithms – Hill-climbing search, Simulated annealing, Genetic Algorithm, Adversarial Search: Game Trees and Minimax Evaluation, Elementary two-players games: tic-tac-toe, Minimax with Alpha-Beta Pruning. | | |
| **Module:4** | **Logic and Reasoning** | **8 hours** |
| Introduction to Logic and Reasoning -Propositional Logic-First Order Logic-Inference in First Order Logic- Unification, Forward Chaining, Backward Chaining, Resolution. | | |
| **Module:5** | **Uncertain Knowledge and Reasoning** | **5 hours** |
| Quantifying Uncertainty- Bayes Rule -Bayesian Belief Network- Approximate Inference in Bayesian networks | | |
| **Module:6** | **Planning** | **7 hours** |
| Classical planning, Planning as State-space search, Forward search, backward search, Planning graphs, Hierarchical Planning, Planning and acting in Nondeterministic domains – Sensor-less Planning, Multiagent planning | | |
| **Module:7** | **Communicating, Perceiving and Acting** | **6 hours** |
| Communication-Fundamentals of Language -Probabilistic Language Processing -Information Retrieval- Information Extraction-Perception-Image Formation- Object Recognition. | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | | |
| | **Total Lecture hours:** | **45 hours** |

**Text Book**

| | |
|---|---|
| 1. | Russell, S. and Norvig, P. 2015. Artificial Intelligence - A Modern Approach, 3rd Edition, Prentice Hall. |

| **Reference Books** | | | |
|---|---|---|---|
| 1. | K. R. Chowdhary, Fundamentals of Artificial Intelligence, Springer, 2020. | | |
| 2 | Alpaydin, E. 2010. Introduction to Machine Learning. 2nd Edition, MIT Press. | | |
| Mode of Evaluation: CAT, Assignment, Quiz, FAT | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE307L | Compiler Design | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To provide fundamental knowledge of various language translators.
2. To make students familiar with lexical analysis and parsing techniques.
3. To understand the various actions carried out in semantic analysis.
4. To make the students get familiar with how the intermediate code is generated.
5. To understand the principles of code optimization techniques and code generation.
6. To provide foundation for study of high-performance compiler design.

**Course Outcomes**

1. Apply the skills on devising, selecting, and using tools and techniques towards compiler design
2. Develop language specifications using context free grammars (CFG).
3. Apply the ideas, the techniques, and the knowledge acquired for the purpose of developing software systems.
4. Constructing symbol tables and generating intermediate code.
5. Obtain insights on compiler optimization and code generation.

| **Module:1** | **INTRODUCTION TO COMPILATION AND LEXICAL ANALYSIS** | **7 hours** |
|---|---|---|

Introduction to LLVM - Structure and Phases of a Compiler-Design Issues-Patterns-Lexemes-Tokens-Attributes-Specification of Tokens-Extended Regular Expression- Regular expression to Deterministic Finite Automata (Direct method) - Lex - A Lexical Analyzer Generator.

| **Module:2** | **SYNTAX ANALYSIS** | **8 hours** |
|---|---|---|

Role of Parser- Parse Tree - Elimination of Ambiguity – Top Down Parsing - Recursive Descent Parsing - LL (1) Grammars – Shift Reduce Parsers- Operator Precedence Parsing - LR Parsers, Construction of SLR Parser Tables and Parsing- CLR Parsing- LALR Parsing.

| **Module:3** | **SEMANTICS ANALYSIS** | **5 hours** |
|---|---|---|

Syntax Directed Definition – Evaluation Order - Applications of Syntax Directed Translation - Syntax Directed Translation Schemes - Implementation of L-attributed Syntax Directed Definition.

| **Module:4** | **INTERMEDIATE CODE GENERATION** | **5 hours** |
|---|---|---|

Variants of Syntax trees - Three Address Code- Types – Declarations - Procedures - Assignment Statements - Translation of Expressions - Control Flow - Back Patching- Switch Case Statements.

| **Module:5** | **CODE OPTIMIZATION** | **6 hours** |
|---|---|---|

Loop optimizations- Principal Sources of Optimization -Introduction to Data Flow Analysis - Basic Blocks - Optimization of Basic Blocks - Peephole Optimization- The DAG Representation of Basic Blocks -Loops in Flow Graphs - Machine Independent Optimization- Implementation of a naïve code generator for a virtual Machine- Security checking of virtual machine code.

| **Module:6** | **CODE GENERATION** | **5 hours** |
|---|---|---|

Issues in the design of a code generator- Target Machine- Next-Use Information - Register Allocation and Assignment- Runtime Organization- Activation Records.

| **Module:7** | **PARALLELISM** | **7 hours** |
|---|---|---|

Parallelization- Automatic Parallelization- Optimizations for Cache Locality and Vectorization- Domain Specific Languages-Compilation- Instruction Scheduling and Software Pipelining- Impact of Language Design and Architecture Evolution on Compilers-Static Single Assignment

| **Module:8** | **Contemporary Issues** | **2 hours** |
|---|---|---|

| | | |
|---|---|---|
| | **Total Lecture hours:** | **45 hours** |
| **Text Book(s)** | | |
| 1. | A. V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, Compilers: Principles, techniques, & tools, 2007, Second Edition, Pearson Education, Boston. | |
| **Reference Books** | | |
| 1. | Watson, Des. A Practical Approach to Compiler Construction. Germany, Springer International Publishing, 2017. | |
| Mode of Evaluation: CAT, Quiz, Written assignment and FAT | | |
| Recommended by Board of Studies | 04-03-2022 | |
| Approved by Academic Council | No. 65 | Date 17-03-2022 |

| BCSE307P | Compiler Design Lab | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To provide fundamental knowledge of various language translators.
2. To make students familiar with phases of compiler.
3. To provide foundation for study of high-performance compiler design.

**Course Outcome**

1. Apply the skills on devising, selecting and using tools and techniques towards compiler design
2. Develop language specifications using context free grammars (CFG).
3. Apply the ideas, the techniques, and the knowledge acquired for the purpose of developing software systems.
4. Constructing symbol tables and generating intermediate code.
5. Obtain insights on compiler optimization and code generation.

| **Indicative Experiments** | |
|---|---|
| 1. | Implementation of LEXR using LLVM. |
| 2. | Implementation of handwritten parser using LLVM |
| 3. | Generating code with the LLVM backend. |
| 4. | Defining a real programming language. |
| 5. | Write a recursive descent parser for the CFG language and implement it using LLVM. |
| 6. | Write a LR parser for the CFG language and implement it in the using LLVM. |
| 7. | Intro to Flex and Bison<br>Modify the scanner and parser so that terminating a statement with "; b" instead of ";" results in the output being printed in binary. |
| 8. | Using LLVM-style RTTI for the AST and Generating IR from the AST. |
| 9. | Converting types from an AST description to LLVM types. |
| 10. | Emitting assembler text and object code. |
| | **Total Laboratory Hours** \| **30 hours** |

Mode of assessment: CAT, FAT

**Text Book(s)**

| | |
|---|---|
| 1 | Learn LLVM 12: A beginner's guide to learning LLVM compiler tools and core libraries with C++ |

**Reference Books**

| | |
|---|---|
| 1. | Watson, Des. A Practical Approach to Compiler Construction. Germany, Springer International Publishing, 2017. |

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE308L | Computer Networks | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To build an understanding among students about the fundamental concepts of computer networking, protocols, architectures, and applications.
2. To help students to acquire knowledge in design, implement and analyze performance of OSI and TCP-IP based Architectures.
3. To identify the suitable application layer protocols for specific applications and its respective security mechanisms.

**Course Outcomes**

On completion of this course, student should be able to:

1. Interpret the different building blocks of Communication network and its architecture.
2. Contrast different types of switching networks and analyze the performance of network
3. Identify and analyze error and flow control mechanisms in data link layer.
4. Design sub-netting and analyze the performance of network layer with various routing protocols.
5. Compare various congestion control mechanisms and identify appropriate transport layer protocol for real time applications with appropriate security mechanism.

| **Module:1** | **Networking Principles and Layered Architecture** | **6 hours** |
|---|---|---|
| Data Communications and Networking: A Communications Model – Data Communications - Evolution of network, Requirements , Applications, Network Topology (Line configuration, Data Flow), Protocols and Standards, Network Models (OSI, TCP/IP) | | |
| **Module:2** | **Circuit and Packet Switching** | **7 hours** |
| Switched Communications Networks – Circuit Switching – Packet Switching – Comparison of Circuit Switching and Packet Switching – Implementing Network Software, Networking Parameters(Transmission Impairment, Data Rate and Performance) | | |
| **Module:3** | **Data Link Layer** | **8 hours** |
| Error Detection and Correction – Hamming Code , CRC, Checksum- Flow control mechanism – Sliding Window Protocol - GoBack - N - Selective Repeat - Multiple access Aloha - Slotted Aloha - CSMA, CSMA/CD – IEEE Standards(IEEE802.3 (Ethernet), IEEE802.11(WLAN))- RFID- Bluetooth Standards | | |
| **Module:4** | **Network Layer** | **8 hours** |
| IPV4 Address Space – Notations – Classful Addressing – Classless Addressing – Network Address Translation – IPv6 Address Structure – IPv4 and IPv6 header format | | |
| **Module:5** | **Routing Protocols** | **6 hours** |
| Routing-Link State and Distance Vector Routing Protocols- Implementation-Performance Analysis- Packet Tracer | | |
| **Module:6** | **Transport Layer** | **5 hours** |
| TCP and UDP-Congestion Control-Effects of Congestion-Traffic Management-TCP Congestion Control-Congestion Avoidance Mechanisms-Queuing Mechanisms-QoS Parameters | | |
| **Module:7** | **Application layer** | **3 hours** |
| Application layer-Domain Name System-Case Study : FTP-HTTP-SMTP-SNMP | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | | |
| | **Total Lecture hours:** | **45 hours** |

**Text Book**

1. Behrouz A. Forouzan, Data communication and Networking, 5th Edition, 2017,

| | | | |
|---|---|---|---|
| | McGraw Hill Education. | | |
| **Reference Books** | | | |
| 1. | James F. Kurose and Keith W.Ross, Computer Networking: A Top-Down Approach, 6th Edition, 2017, Pearson Education. | | |
| 2. | William Stallings, "Data and Computer Communication", 10th Edition, 2017, Pearson, United Kingdom. | | |
| **Mode of Evaluation**: CAT, Written Assignment, Quiz, FAT | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE308P | Computer Networks Lab | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To build an understanding among students about the fundamental concepts of computer networking, protocols, architectures, and applications.
2. To help students to acquire knowledge in design, implement and analyze performance of OSI and TCP-IP based Architectures.
3. To identify the suitable application layer protocols for specific applications and its respective security mechanisms

**Course Outcome**

On completion of this course, student should be able to:

1. Interpret the different building blocks of Communication network and its architecture.
2. Contrast different types of switching networks and analyze the performance of network
3. Identify and analyze error and flow control mechanisms in data link layer.
4. Design sub-netting and analyze the performance of network layer with various routing protocols.
5. Compare various congestion control mechanisms and identify appropriate transport layer protocol for real time applications with appropriate security mechanism.

**Indicative Experiments**

| | |
|---|---|
| 1. | Study of Basic Network Commands, Demo session of all networking hardware and Functionalities |
| 2. | Error detection and correction mechanisms |
| 3. | Flow control mechanisms |
| 4. | IP addressing Classless addressing |
| 5. | Observing Packets across the network and Performance Analysis of Routing protocols |
| 6. | Socket programming(TCP and UDP) - Some challenging experiments can be given on Socket programming |
| 7. | Simulation of unicast routing protocols |
| 8. | Simulation of Transport layer Protocols and analysis of congestion control techniques in network |
| 9. | Develop a DNS client server to resolve the given host name or IP address |
| | **Total Laboratory Hours** 30 hours |

**Text book**

1 W.Richard Stevens, Uix Network Programming, 2ndEdition, Pearson Education, 2015.

**Mode of assessment**: Continuous assessment, FAT

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE309L | Cryptography and Network Security | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To explore the concepts of basic number theory and cryptographic techniques.
2. To impart concept of Hash and Message Authentication, Digital Signatures and authentication protocols.
3. To reveal the basics of transport layer security, Web Security and various types of System Security.

**Course Outcomes**

On completion of this course, students should be able to:

1. To know the fundamental mathematical concepts related to security.
2. To understand concept of various cryptographic techniques.
3. To apprehend the authentication and integrity process of data for various applications
4. To know fundamentals of Transport layer security, web security, E-Mail Security and IP Security

| | | |
|---|---|---|
| **Module:1** | **Fundamentals of Number Theory** | **5 hours** |
| Finite Fields and Number Theory: Modular arithmetic, Euclidian Algorithm, Primality Testing: Fermats and Eulers theorem, Chinese Reminder theorem, Discrete Logarithms. | | |
| **Module:2** | **Symmetric Encryption Algorithms** | **7 hours** |
| Symmetric key cryptographic techniques: Introduction to Stream cipher, Block cipher: DES, AES,IDEA, Block Cipher Operation, Random Bit Generation and RC4 | | |
| **Module:3** | **Asymmetric Encryption Algorithm and Key Exchange** | **8 hours** |
| Asymmetric key cryptographic techniques: principles, RSA, ElGamal, Elliptic Curve cryptography, Homomorphic Encryption and Secret Sharing, Key distribution and Key exchange protocols, Diffie-Hellman Key Exchange, Man-in-the-Meddle Attack | | |
| **Module:4** | **Message Digest and Hash Functions** | **5 hours** |
| Requirements for Hash Functions, Security of Hash Functions, Message Digest (MD5), Secure Hash Function (SHA),Birthday Attack, HMAC | | |
| **Module:5** | **Digital Signature and Authentication Protocols** | **7 hours** |
| Authentication Requirements, Authentication Functions, Message Authentication Codes, Digital Signature Authentication, Authentication Protocols, Digital Signature Standards, RSA Digital Signature, Elgamal based Digital Signature, Authentication Applications: Kerberos, X.509 Authentication Service, Public Key Infrastructure (PKI) | | |
| **Module:6** | **Transport Layer Security and IP Security** | **4 hours** |
| Transport-Layer Security, Secure Socket Layer(SSL),TLS, IP Security: Overview: IP Security Architecture, Encapsulating Payload Security | | |
| **Module:7** | **E-mail, Web and System Security** | **7 hours** |
| Electronic Mail Security, Pretty Good Privacy (PGP), S/MIME, Web Security: Web Security Considerations, Secure Electronic Transaction Protocol<br>Intruders, Intrusion Detection, Password Management, Firewalls: Firewall Design Principles, Trusted Systems. | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | | |
| | **Total Lecture hours:** | **45 hours** |

**Text Book**

1. Cryptography and Network Security-Principles and Practice, 8th Edition, by Stallings

| | | | |
|---|---|---|---|
| | William, published by Pearson, 2020 | | |
| **Reference Books** | | | |
| 1. | Cryptography and Network Security, 3rd Edition, by Behrouz A Forouzan and Depdeep Mukhopadhyay, published by McGrawHill, 2015 | | |
| **Mode of Evaluation**: CAT, written assignment, Quiz, and FAT | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE309P | Cryptography and Network Security Lab | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. Understand various Private and Public Key cryptographic algorithms.
2. To learn about hash functions and digital signature algorithms
3. Acquire knowledge in various network security models

**Course Outcome**

On completion of this course, students should be able to:

1. Implement various cipher techniques without using standard cryptographic library functions
2. Develop the various hash functions and digital signature algorithms for different applications
3. Develop various secured networking-based application

**Indicative Experiments**

| | |
|---|---|
| 1. | Consider a sender and receiver who need to exchange data confidentially using symmetric encryption. Write program that implements DES encryption and decryption using a 64 bit key size and 64 bit block size |
| 2. | Consider a sender and receiver who need to exchange data confidentially using symmetric encryption. Write program that implements AES encryption and decryption using a 64/128/256 bits key size and 64 bit block size. |
| 3 | Develop an chipper scheme by using RSA |
| 4. | Develop a MD5 hash algorithm that finds the Message Authentication Code (MAC) |
| 5 | Find a Message Authentication Code (MAC) for given variable size message by using SHA-128 and SHA-256 Hash algorithm<br>Measure the Time consumptions for varying message size for both SHA-128 and SHA-256. |
| 6 | Develop the Digital Siganture standard(DSS)for verifying the legal communicating parties |
| 7 | Design a Diffie Hellman multiparty key exchange protocol and perform Man-in-the-Middle Attack. |
| 8 | Develop a simple client and server application using SSL socket communication |
| 9 | Develop a simple client server model using telnet and capture the packets transmitted with tshark Analyze the pcap file and get the transmitted data (plain text) using any packet capturing library.<br>Implement the above scenario using SSH and observe the data |
| 10 | Develop a web application that implements JSON web token |
| | **Total Laboratory Hours** 30 hours |

**Mode of assessment**: Continuous Assessment, FAT

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE324L | FOUNDATIONS OF BLOCKCHAIN TECHNOLOGY | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To understand building blocks of Blockchain.
2. To significance of Distributed Ledger Technology and Smart Contract.
3. To exploit applications of Blockchain in real world scenarios and their impacts.

**Course Outcomes**

After completion of this course, the student shall be able to:

1. Understand Blockchain ecosystem and its services in real world sceneries
2. Apply and Analyze the requirement of Distributed Ledger Technology and Smart Contract
3. Design and Demonstrate end-to-end decentralized applications
4. Acquaint the protocol and assess their computational requirements

| **Module:1** | **Foundations of Blockchain** | **7 hours** |
|---|---|---|

Blockchain Architecture – Challenges – Applications – Blockchain Design Principles -The Blockchain Ecosystem - The consensus problem - Asynchronous Byzantine Agreement - AAP protocol and its analysis - peer-to-peer network – Abstract Models - GARAY model - RLA Model - Proof of Work (PoW) - Proof of Stake (PoS) based Chains - Hybrid models.

| **Module:2** | **Distributed Ledger Technology** | **6 hours** |
|---|---|---|

Origin of Ledgers – Types and Features of Distributed Ledger Technology (DLT) - Role of Consensus Mechanism - DLT Ecosystem - Distributed Ledger Implementations – Blockchain - Ethereum - Public and Private Ledgers – Registries – Ledgers - Practitioner Perspective: Keyless Technologies, Transparency as a Strategic Risk, Transparency as a Strategic Asset, Usage of Multiple IDs - Zero Knowledge Proofs - Implementation of Public and Private Blockchain

| **Module:3** | **Smart Contracts** | **5 hours** |
|---|---|---|

Anatomy of a Smart Contracts - Life Cycle - Usage Patterns - DLT-based smart contracts - Use Cases: Healthcare Industry and Property Transfer.

| **Module:4** | **Decentralized Organization** | **5 hours** |
|---|---|---|

Decentralization versus Distribution - Centralized-distributed (Ce-Di) organizations - Decentralized-distributed (De-Di) organizations - Decentralized Autonomous Organizations: Aragon, DAOstack, DAOhaus and Colony.

| **Module:5** | **Types of Blockchain Ecosystem** | **7 hours** |
|---|---|---|

One-Leader Ecosystem - Joint Venture or Consortia Ecosystems - Regulatory Blockchain Ecosystems - Components in Blockchain Ecosystem: Leaders, Core Group, Active Participants, Users, Third-Party Service Providers - Governance for Blockchain Ecosystems.

| **Module:6** | **Blockchain Protocols** | **6 hours** |
|---|---|---|

Ethereum tokens – Augur - Golem - Understanding Ethereum tokens - App Coins and Protocol Tokens - Blockchain Token Securities Law Framework - Token Economy - Token sale structure - Ethereum Subreddit.

| **Module:7** | **High Performance Computing** | **7 hours** |
|---|---|---|

Integrity of High Performance Systems - Data Provenance - Cluster Construction and Deployment - Mock Workload - Blockchain Software Evaluation - Blockchain storage of Integrity Data.

| **Module:8** | **Contemporary Issues** | **2 hours** |
|---|---|---|
| | **Total Lecture hours:** | **45 hours** |

**Text Book**

1. Dhillon, V., Metcalf, D., and Hooper, M, Blockchain enabled applications, 2017, 1st

| | | | |
|---|---|---|---|
| | Edition, CA: Apress, Berkeley. | | |
| | | | |
| **Reference Books** | | | |
| 1. | Diedrich, H., Ethereum: Blockchains, digital assets, smart contracts, decentralized autonomous organizations, 2016, 1st Edition, Wildfire publishing, Sydney. | | |
| 2. | Wattenhofer, R. P, Distributed Ledger Technology: The Science of the Blockchain (Inverted Forest Publishing), 2017, 2nd Edition, Createspace Independent Pub, Scotts Valley, California, US. | | |
| Mode of Evaluation: CAT, written assignment, Quiz, FAT | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE325L | INTRODUCTION TO BITCOIN | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To Identify the process of Cryptocurrency.
2. To understand the functionality of Bitcoin.
3. To explore the recent developments on Bitcoin.

**Course Outcomes**

After completion of this course, the student shall be able to:

1. Understand the fundamentals of Cryptography.
2. Gain knowledge about various operations associated with Cryptocurrency.
3. Develop the methods for verification and validation of Bitcoin transactions.
4. Apply the principles, practices and policies associated with Bitcoin business.

| | | |
|---|---|---|
| **Module:1** | **Fundamentals of Cryptography** | **5 hours** |
| Cryptographic Hash Functions - Hash Pointers and Data Structures - Digital Signatures - Public Keys as Identities - A Simple Cryptocurrency. | | |
| **Module:2** | **Features of Bitcoin** | **6 hours** |
| Bitcoin Transactions - Bitcoin Scripts - Applications of Bitcoin Scripts - Bitcoin Blocks - Bitcoin Network and Limitations. | | |
| **Module:3** | **Bitcoin Techniques** | **7 hours** |
| Techniques to Store and Use Bitcoins - Hot and Cold Storage - Splitting and Sharing Keys - Online Wallets and Exchanges - Payment Services - Transaction Fees - Bitcoin Trading. | | |
| **Module:4** | **Bitcoin Mining** | **8 hours** |
| Task of Bitcoin Miners - Mining Hardware - Energy Consumption and Ecology - Mining Pools - Mining Incentives - Merkley Tree - hardness of mining - transaction verifiability. | | |
| **Module:5** | **Bitcoin and Anonymity** | **5 hours** |
| Anonymity – Re-identification of Bitcoin - Mixing and Decentralisation of Bitcoin - Zero coin and Zero cash. | | |
| **Module:6** | **Mining Strategies** | **5 hours** |
| Essential Puzzle Requirements – Application Specific Integrated Circuit Resistant(ASIC) Puzzles - Proof of Volunteer computing - Non externalization of Puzzles - Proof of Stake Virtual Mining. | | |
| **Module:7** | **Bitcoin as a Platform** | **7 hours** |
| Bitcoin as an Append-Only Log - Bitcoin as Smart Property - Secure Multi-Party Lotteries in Bitcoin - Bitcoin as Randomness Source - Prediction Markets and Real-World Data Feeds. | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | **Total Lecture hours:** | **45 hours** |

**Text Book**

1. Goldfeder, S., Bonneau, J., Miller, A., Felten, E., Narayanan, A. Bitcoin and Cryptocurrency Technologies, 2016, 1st edition, Princeton University Press, New Jersey.

**Reference Books**

1. Antonopoulos, A. M. Mastering Bitcoin: unlocking digital cryptocurrencies, 2017, 2nd edition, OReilly Media, Inc, United States.
2. Lewis, Antony, The Basics Of Bitcoins and Blockchains: An Introduction To Cryptocurrencies and The Technology That Powers Them., 2018, 1st edition, Mango Media Inc., United States.

Mode of Evaluation: CAT / Assignment / Quiz / FAT

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE326L | BLOCKCHAIN ARCHITECTURE DESIGN | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To provide the knowledge on Blockchain architecture.
2. To understand the design of Blockchain transaction and security issues.
3. To study about various use Cases in Blockchain.

**Course Outcome**

After completion of this course, the student shall be able to:

1. Understand the requirements of the fundamentals of Blockchain.
2. Identify and apply the concept of Bitcoin.
3. Recognize the underlying technology of transactions, blocks and proof-of-work.
4. Gain a deep insight into Bitcoin network, Bitcoin miners and Bitcoin transactions.
5. Design and explore the applications of Blockchain.

| **Module:1** | **Fundamentals of Blockchain** | **6 hours** |
|---|---|---|

Blockchain: Importance and features – Layers of Blockchain: application layer, execution layer, semantic layer, propagation layer, consensus layer – Types of Blockchain – Blockchain in practical use today – Blockchain governance challenges – Blockchain technical challenges.

| **Module:2** | **Blockchain for Enterprise** | **6 hours** |
|---|---|---|

Blockchain Components and Concepts - Block Header and Identifiers - Linking Blocks in the Blockchain - Mining and Consensus: Aggregating transactions into Blocks - Mining the Block - Validating and Assembling of Blocks, Selecting Chains of Blocks.

| **Module:3** | **Transactions and Bitcoin Network** | **6 hours** |
|---|---|---|

Transactions: Lifecycle, Structure, Inputs and Outputs, Standard Transactions - Bitcoin Network: Network discovery for a new node, Block propagation.

| **Module:4** | **Bitcoin Client** | **8 hours** |
|---|---|---|

Consensus in Bitcoin: Proof of Work (PoW), Mining the Block, Changing the Consensus Rules - Bitcoin Core: Bitcoin core application programming interface, running a bitcoin core node, Alternative clients, libraries and toolkits - Bitcoin Addresses: Implementing Keys and Addresses in Python – Wallets.

| **Module:5** | **Security and privacy practices** | **6 hours** |
|---|---|---|

Security Architecture principles - Technical and inherent risks of the blockchain technology - Attacks on Privacy: Blockchain and non-blockchain based Attacks - Risks and Limitations of Blockchain – User security best practices: physical bitcoin storage, hardware wallets, balancing risk, diversifying risk, multi signature and governance.

| **Module:6** | **Blockchain Architecture and Applications** | **6 hours** |
|---|---|---|

Design methodology for blockchain applications: blockchain application templates, blockchain application development – Ethereum – Solidity - Deploying a sample application: Blockchain and betting – Colored coins – Counterparty.

| **Module:7** | **Blockchain Use Cases** | **5 hours** |
|---|---|---|

Blockchain in Financial Software and Systems - Supply chain and logistics monitoring - Music royalties tracking - Advertising insights - Blockchain implementation for Land Records - Digital content publishing and selling - Digital Supply chain - Medical Record Management System

| **Module:8** | **Contemporary Issues** | **2 hours** |
|---|---|---|
| | **Total Lecture hours:** | **45 hours** |

**Text Book(s)**


1. Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, Beginning Blockchain, A Beginner's Guide to Building Blockchain Solutions, 2018, 1st edition, Apress, New York.
2. Joseph J. Bambara, Paul R. Allen, Blockchain: a practical guide to developing business,

| | | | |
|---|---|---|---|
| | law and technology solutions, 2018, 1st edition, McGraw-Hill publication, New York. | | |
| **Reference Books** | | | |
| 1. | Swan Melanie, Blockchain: Blueprint for a new economy, 2015, 1st edition, O'Reilly Media, United States. | | |
| Mode of Evaluation: CAT / written assignment / Quiz / FAT | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE327L | SMART CONTRACTS | L | T | P | C |
|---|---|---|---|---|---|
| | | 2 | 0 | 0 | 2 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To understand the Smart Contracts in Blockchain.
2. To learn the tools and programming skills required to generate Smart Contracts.
3. To assess the efficiency of the security issues.

**Course Outcomes**

After completion of this course, the student shall be able to:

1. Understand the basics and objectives of Smart Contracts in a Blockchain.
2. Evaluate the various functionalities and features in an Ethereum to generate Smart Contracts.
3. Introduce the Solidity language in creation of a Smart Contracts.
4. Incorporate Smart Contracts in decentralized applications.
5. Assess the security issues and effectiveness of a Smart Contracts in real world scenarios.

| | | |
|---|---|---|
| **Module:1** | **Fundamentals of Smart Contracts** | **2 hours** |
| Blockchain Terminologies - Cryptocurrency and Smart Contracts - Understanding the Virtual Machine of a Blockchain - Terminology, concepts and practices in Smart Contracts. | | |
| **Module:2** | **Ethereum Smart Contracts** | **5 hours** |
| Definition of Ethereum - Prevalence of the Ethereum blockchain in Smart Contracts development - Ethereum Virtual Machine (EVM) - Instances of working Ethereum Smart Contracts. | | |
| **Module:3** | **Various Aspects in Application of Smart Contracts** | **5 hours** |
| Market impact and scientific innovation – Trust - Security, using Merkle Trees - Future-resistance features in Smart Contracts applications - Workflow of developing a Smart Contracts - Execution environments in writing a Smart Contracts. | | |
| **Module:4** | **Solidity Language Basics** | **4 hours** |
| Layout of a Solidity Source File - Structure of a contracts - Control structures – Functions - Scoping and declarations. | | |
| **Module:5** | **Solidity with Contracts** | **4 hours** |
| Creating contracts - Object-oriented high level language features - Visibility and Getters – Events - Abstract Contracts. | | |
| **Module:6** | **Decentralized Applications** | **4 hours** |
| Decentralized Application Architecture - Connecting to the Blockchain and Smart Contracts – Building dApps – Deployment. | | |
| **Module:7** | **Security Issues** | **4 hours** |
| Shifting from Trust-in-People to Trust-in-Code - Data permanence - Selective-Obscurity - Security counter measures. | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | **Total Lecture hours:** | **30 hours** |

**Text Book**

1. Gavin Zheng, Longxiang Gao, Liqun Huang, Jian Guan, Ethereum Smart Contracts Development in Solidity, 2021, 1st Edition, Springer Singapore.

**Reference Books**

1. Dannen, C., Introducing Ethereum and solidity, 2017, (Vol. 318). Berkeley: Springer.
2. Modi, Ritesh, Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and Blockchain, 2018, Packt Publishing Ltd, United Kingdom.
3. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder,

| | | | |
|---|---|---|---|
| | Bitcoin and cryptocurrency technologies: a comprehensive introduction, 2016, Princeton University Press. | | |
| Mode of Evaluation: CAT / written assignment / Quiz / FAT | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE327P | SMART CONTRACTS LAB | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To understand the Smart Contracts in Blockchain.
2. To learn the tools and programming skills required to generate Smart Contracts.
3. To assess the efficiency of the security issues.

**Course Outcomes**

After completion of this course, the student shall be able to:

1. Evaluate the various functionalities and features in an Ethereum to generate Smart Contracts.
2. Assess the security issues and effectiveness of a Smart Contracts in real world scenarios.

**Indicative Experiments**

1. Setting up Ethereum network by using Geth command line interface.
2. Identifying and setting up a testnet, like Ropsten or Kovan, so that free ethers can be used as transaction.
3. Transfer ethers from one account to another on an Ethereum testnet.
4. Constructing Solidity code for a decentralized application where the owner can create a contracts (with a tenant) which can be replicated to all nodes.
5. In a rented house setup with the owner and the tenants, the tenant can submit a deposit and the contracts's state changes on all the decentralized nodes.
6. The owner should be able to check the balance of the contracts from any one of the nodes.
7. Using Remix on the Solidity code to develop, compile and deploy the contracts.
8. Using setter and getter functions to interact with the contracts
9. Withdrawing funds from a contracts to a restricted account, preferably the owner's, with different levels of security restrictions.
10. Deploying a contracts on an external blockchain by using Ganache and/or MyEtherwalllet, Metamask.

| **Total Laboratory Hours** | **30 hours** |
|---|---|

**Text Book**

1. Gavin Zheng, Longxiang Gao, Liqun Huang, Jian Guan, Ethereum Smart Contracts Development in Solidity, 2021, 1st Edition, Springer Singapore.

**Reference Books**

1. Modi, Ritesh. Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain. 2018, Packt Publishing Ltd, United Kingdom.
2. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Bitcoin and cryptocurrency technologies: a comprehensive introduction, 2016, Princeton University Press.

Mode of assessment: Continuous assessment / FAT

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE328L | CRYPTOCURRENCY TECHNOLOGIES | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To introduce the cryptocurrency concepts and techniques used in business transactions.
2. To provide skills and knowledge about operations and management in cryptocurrency technologies applied in large scale business.
3. To develop own cryptocurrencies that meets the business and customer needs.

**Course Outcome**

After completion of this course, the student shall be able to:

1. Understand the evolution, principles and benefits of cryptocurrencies.
2. Assess existing technologies to choose an appropriate technology that meets business needs.
3. Implement the scripting foundations to cater the needs of generating own cryptocurrencies.
4. Decide a suitable model to capture the business needs by interpreting different crypto primitives.
5. Infer the various bitcoin related security and privacy issues and building own cryptocurrencies.

| **Module:1** | **Fundamentals of Cryptocurrency** | **7 hours** |
|---|---|---|

Cryptocurrency - Origin and Importance - Legal Status - Usage of Cryptocurrency - Blockchain Structure - Interaction between Blockchain and Cryptocurrencies - Importance and uses of Cryptocurrency - Hardware and Software requirements of Block chain.

| **Module:2** | **Functional Aspects of Cryptocurrency** | **8 hours** |
|---|---|---|

Bitcoin and other Cryptocurrencies - Distributed consensus and atomic broadcast - Alternatives to Bitcoin consensus - Alternative coins - Byzantine fault-tolerant consensus methods - Blockchain based cryptocurrency and its applications - Technologies borrowed in Blockchain.

| **Module:3** | **Bitcoin Scripting** | **5 hours** |
|---|---|---|

Bitcoin scripting language and their uses - Transactions - Signatures - Pay to script hash - Segregated Witness - Pay To Multi-signature - Storing Data - Timelocks - Hash Time-Locked Contracts - Atomic Swaps - Payment Channels.

| **Module:4** | **Crypto Primitives for Cryptocurrency** | **5 hours** |
|---|---|---|

Hash functions - Puzzle-friendly Hash - Collison resistant hash - Hash pointers and digital signatures - public key crypto - verifiable random functions - Zero-knowledge systems - Bitcoin Blockchain - Interaction with the blockchain - Elliptic curve cryptography in blockchain - SHA-256.

| **Module:5** | **Security & Privacy Issues in Cryptocurrency** | **4 hours** |
|---|---|---|

Building a Secure Bitcoin payment system - Building a Secure payment gateway - Compiling Bitcoin from source new cryptocurrency - Cloning Bitcoin - Reader coin rebranding - Securing Peer-to-Peer Auctions in Ethereum - Applications of blockchain in cyber security.

| **Module:6** | **Building Own Cryptocurrency** | **7 hours** |
|---|---|---|

Coding Own Cryptocurrency on Ethereum - Building ERC-20 Token - Integrity of information - E-Governance and other contract enforcement mechanisms - Limitations of blockchain - Myths vs. reality of blockchain technology.

| **Module:7** | **Future Directions of Cryptocurrency** | **7 hours** |
|---|---|---|

| | | |
|---|---|---|
| Smart Property - Efficient micro-payments - Coupling Transactions and Payment (Interdependent Transactions) - Public Randomness Source Prediction Markets - Escrow transactions - Green addresses - Auctions and Markets - Multi-party Lotteries. | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | | |
| | **Total Lecture hours:** | **45 hours** |

| | |
|---|---|
| **Text Book** | |
| 1. | Daskalakis, Nikos, and Panagiotis Georgitseas. An Introduction to Cryptocurrencies: The Crypto Market Ecosystem, 2020, 1st Edition, Routledge, New York. |
| **Reference Books** | |
| 1. | Grabowski, Mark. Cryptocurrencies: A Primer on Digital Money, 2019, 1st Edition, Routledge, New York. |
| 2. | Narayanan, Arvind, et al. Bitcoin and cryptocurrency technologies: a comprehensive introduction, 2016, 1st Edition, Princeton University Press, New Jersey |

| | | | |
|---|---|---|---|
| Mode of Evaluation: CAT / written assignment / Quiz / FA | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Counc | No. 65 | Date | 17-03-2022 |

| BCSE329L | BLOCKCHAIN AND DISTRIBUTED LEDGER TECHNOLOGY | L | T | P | C |
|---|---|---|---|---|---|
| | | 2 | 0 | 0 | 2 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To understand Blockchain and Distributed Ledger Technologies.
2. To learn the development in Blockchain functionalities.
3. To identify alternative techniques to proof of work for Blockchain protocols, proof of stake/space.

**Course Outcomes**

After completion of this course, the student shall be able to:

1. Comprehend the functionality of blockchain.
2. Choose a blockchain implementation based on real time scenario.
3. Examine the techniques for anonymity preservation.
4. Determine the Blockchain challenges.
5. Identify the use cases of distributed ledger technology.
6. Evaluate alternative blockchain and their applicability.

| **Module:1** | **Blockchain and Distributed Ledger Fundamentals** | **4 hours** |
|---|---|---|

Blockchain - Distributed Ledger - Cryptographic basics for cryptocurrency - signature schemes, encryption schemes and elliptic curve cryptography - CAP theorem - Categories of Blockchain: Public blockchain, Private blockchain, Permissioned Ledger, Tokenized blockchain, Tokenless blockchain, and Sidechains.

| **Module:2** | **Blockchain Functionality** | **5 hours** |
|---|---|---|

Distributed identity: Public and private keys, Digital identification and wallets - Decentralized network - Permissioned distributed Ledger - Blockchain data structure - Double spending - Network consensus - Sybil attacks - Block rewards and miners - Forks and consensus chain - Finality in Blockchain Consensus - Limitation of proof-of-work - Alternatives to Proof of Work.

| **Module:3** | **Blockchain Implementation** | **4 hours** |
|---|---|---|

Bitcoin and Merkle Root - Eventual Consistency and Bitcoin - Byzantine Fault Tolerance - Bitcoin and Secure Hashing - Bitcoin block-size - Bitcoin Mining - Blockchain Collaborative Implementations: Hyperledger, Corda - Ethereum's ERC 20 and token explosion.

| **Module:4** | **Decentralization using Blockchain** | **4 hours** |
|---|---|---|

Blockchain and full ecosystem decentralization: Smart contract, Decentralized autonomous organization (DAO), Decentralized applications - Platforms for decentralization.

| **Module:5** | **Zero Knowledge Proofs and Protocols in Blockchain** | **4 hours** |
|---|---|---|

Pseudo-anonymity vs. anonymity - Succinct non interactive argument for Knowledge (SNARK) - pairing on Elliptic curves – Zcash - Zk-SNARKS for anonymity preservation.

| **Module:6** | **Blockchain Challenges** | **3 hours** |
|---|---|---|

Blockchain Governance Challenges: Bitcoin Blocksize Debate, The Ethereum DAO Fork, Ethereum's Move to PoS and Scaling Challenges - Blockchain Technical Challenges: Denial-of-Service Attacks, Security in Smart Contracts, Scaling, Sharding.

| **Module:7** | **Distributed Ledger Technology in Alternative Blockchain** | **4 hours** |
|---|---|---|

Kadena, Ripple, Stellar, Rootstock, Drivechain, Quorum – Decentralized Network manager: Tezos, Maidsafe, BigChainDB - Decentralized Cloud Storage: Storj.

| **Module:8** | **Contemporary Issues** | **2 hours** |
|---|---|---|
| | **Total Lecture hours:** | **30 hours** |

**Text Book**

1. Goldfeder, S., Bonneau, J., Miller, A., Felten, E., Narayanan, A. Bitcoin and

| | | | | |
|---|---|---|---|---|
| | Cryptocurrency Technologies, 2016, 1st edition, Princeton University Press, New Jersey. | | | |
| **Reference Books** | | | | |
| 1. | Iyer, Kedar, et al. Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions., 2018, 1st edition, McGraw-Hill Education, United Kingdom. | | | |
| 2. | Wattenhofer, R. Distributed Ledger Technology: The Science of the Blockchain, 2017, 1st edition, CreateSpace Independent Publishing Platform, United States. | | | |
| Mode of Evaluation: CAT / written assignment / Quiz / FAT | | | | |
| Recommended by Board of Studies | | 04-03-2022 | | |
| Approved by Academic Council | | No. 65 | Date | 17-03-2022 |

| BCSE329P | BLOCKCHAIN AND DISTRIBUTED LEDGER TECHNOLOGY LAB | L | T | P | C |
|---|---|---|---|---|---|
| | | 0 | 0 | 2 | 1 |
| **Pre-requisite** | **NIL** | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives**

1. To understand Blockchain and Distributed Ledger Technologies.
2. To learn the development in Blockchain functionalities.
3. To identify alternative techniques to proof of work for Blockchain protocols, proof of stake/space.

**Course Outcomes**

After completion of this course, the student shall be able to:

1. Implement a blockchain for real time scenario.
2. Evaluate alternative blockchain and their applicability.

**Indicative Experiments**

| | |
|---|---|
| 1. | Deploy a local private blockchain over a network with Ethereum or Rust. |
| 2. | Implement the **mining** module of Bitcoin client using Rust. The mining module, or miner, should produce blocks that solve proof-of-work puzzle. |
| 3. | Compile and test smart contracts on a testing framework using the Ethereum Virtual Machine (EVM). |
| 4. | Deploy a chaincode using Hyperledger Fabric on a custom network. |
| 5. | Create a Hyperledger Fabric Blockchain service on Cloud. |
| 6. | Deploying a ERC20 token on the Ethereum Testnet. |
| 7. | Launch your own token on alternative blockchain such as BigchainDB |

| | |
|---|---|
| **Total Laboratory Hours** | **30 hours** |

Text Book

| | |
|---|---|
| 1 | Goldfeder, S., Bonneau, J., Miller, A., Felten, E., Narayanan, A. Bitcoin and Cryptocurrency Technologies, 2016, 1st edition, Princeton University Press, New Jersey. |

**Reference Books**

| | |
|---|---|
| 1 | Iyer, Kedar, et al. Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions., 2018, 1st edition, McGraw-Hill Education, United Kingdom. |

Mode of Evaluation: CAT / written assignment / Quiz / FAT

| | | | |
|---|---|---|---|
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |

| BCSE330L | PUBLIC KEY INFRASTRUCTURE AND TRUST MANAGEMENT | L | T | P | C |
|---|---|---|---|---|---|
| | | 3 | 0 | 0 | 3 |
| **Pre-requisite** | | **Syllabus version** | | | |
| | | 1.0 | | | |

**Course Objectives:**

1. To provide the knowledge on Public Key Cryptography techniques and Public Key infrastructure.
2. To study about the Digital Certificates and the security challenges.
3. To understand the various trust models and the trust management systems.

**Course Outcome:**

After completion of this course, the student shall be able to:

1. Analyze and design Public Key cryptographic algorithms.
2. Evaluate the components of PKI and design & integrate PKI services
3. Design the Digital Certificates with PKI considerations
4. Identify the access control mechanism and provide solution for the security challenges
5. Analyze and select suitable trust model and manage with operational considerations

| **Module:1** | **Public Key Cryptography Basics** | **5 hours** |
|---|---|---|

Public Key Cryptography: Secret key, Public key, public/private key pair, Services of public key cryptography - RABIN Cryptosystem - ElGamal Cryptosystem - Message Integrity and Authentication: Random Oracle model, message authentication, Cryptographic hash functions.

| **Module:2** | **Public Key Infrastructure** | **7 hours** |
|---|---|---|

Components and architecture of fully functional Public key infrastructure(PKI): Certification authority, Certificate repository, Certificate revocation, Key backup and recovery, Automatic key update, Key history management, Cross-certification, Support for non-repudiation, Time stamping, Client software, Core PKI Services, PKI-Enabled Services, PKI interoperability, deployment and assessment PKI data structures - PKI architectures: Single CA, Hierarchical PKI, Mesh PKI, Trust Lists, Bridge Certification Authority (CA), Registration Authority (RA), Simple PKI (SPKI), PKI application : Smart card integration with PKI's.

| **Module:3** | **Digital Certificates** | **7 hours** |
|---|---|---|

Introduction to Digital Certificate - Certificate Structure and Semantics - Alternative Certificate Formats - Certificate Policies - Object Identifiers - Policy Authorities - Certification Authority - Key/Certificate Life Cycle Management - Certificate Revocation - Representing certificates in terms of S-Expressions - Certificate Chain.

| **Module:4** | **Access Control Mechanisms and Security Challenges** | **7 hours** |
|---|---|---|

Access Control Mechanisms: Discretionary Access Control (DAC) – Mandatory Access Control (MAC) – Role Based Access Control (RBAC) - Issues : Revocation- Anonymity- Privacy issues - Entity Authentication - Passwords and Challenge Response - zero-knowledge and bio-metrics - Key management - security key distribution – Kerberos - Symmetric Key agreement - Public Key Distribution and Hi-jacking - Issues of revocation - Anonymity and Privacy.

| | | |
|---|---|---|
| **Module:5** | **Trust Models** | **7 hours** |
| Distributed Trust Architecture - Mesh Configuration - Hub-and-Spoke Configuration – Four-Corner Trust Model - Web Model - User-Centric Trust - Cross-Certification - Entity Naming - Certificate Path Processing - Path Construction - Path Validation - Trust Anchor Considerations - Multiple Key Pairs - Key Pair Uses - Relationship between Key Pairs and Certificates. | | |
| **Module:6** | **Trust Management Systems** | **5 hours** |
| Social network based Trust Management System- Reputation based Trust Management System (DMRep, EigenRep, P2Prep) - Framework for Trust Establishment - Risks Impact on E-Commerce and E- Business: Information Risk and Technology Business Risk. | | |
| **Module:7** | **Operational Considerations** | **5 hours** |
| Client-Side Software - Off-line Operations - Physical Security - Hardware Components - User Key Compromise - Disaster Preparation and Recovery - Relying Party Notification – Preparation – Recovery - Electronic Signature Legislation and Considerations. | | |
| **Module:8** | **Contemporary Issues** | **2 hours** |
| | | |
| | **Total Lecture hours:** | **45 hours** |

| | |
|---|---|
| **Text Book(s)** | |
| 1. | John R. Vacca, Public Key Infrastructure: Building Trusted Applications and Web Services, 2019, 1st edition. Auerbach Publications, US. |
| 2. | Carlisle Adams, Steve Lloyd, Understanding PKI: Concepts, Standards, and Deployment Considerations, 2011, 2nd Edition, Addison-Wesley, US. |
| **Reference Books** | |
| 1. | Buchmann J, Karatsiolis E, Wiesmaier A, Karatsiolis E., Introduction to public key infrastructures, 2013, Berlin: Springer. |

| | | | |
|---|---|---|---|
| Mode of Evaluation: CAT / written assignment / Quiz / FAT | | | |
| Recommended by Board of Studies | 04-03-2022 | | |
| Approved by Academic Council | No. 65 | Date | 17-03-2022 |