# B.Sc. Computer Science Syllabus NEP 2020 (Batches 2022 to 2025)

## Fourth Year – NHEQF Level- 6

**In the fourth Year of Four-Year Undergraduate Program (FYUP) two types of programs are offered:**

**1. U.G. with Honours**

**2. U.G. Honours with Research**

Candidates with a minimum CGPA of 7.5 will be eligible to continue their studies into the fourth year of the undergraduate program, leading to a four-year Bachelor's degree (Honours with Research).

## Fourth Year- (U.G. with Honours)

**The following course structure under FYUP for Multidisciplinary programmes is designed for subjects which have practical based courses or have relatively larger emphasis on practical course-based learning.**

**(For practical based subjects)**

| **Entry requirement** | (After completing requirements of a 3-year bachelor's degree (120 credits) and 2 additional credits under SSD, will be allowed to continue studies in the fourth year of the undergraduate programme leading to the four years bachelor's degree (with Honours). | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Course Type** | **Semester-VII** | | | | **Semester-VIII** | | | |
| | **Subject/Title** | **No. of paper** | **Credits** | | **Subject /Title** | **No. of paper** | **Credits** | |
| | | | **T** | **P** | | | **T** | **P** |
| **Major Subject (One)** | **Core Major -I (Computer Organization and Architecture)** | **1** | **5** | **-** | **Core Major -I (Compiler Design)** | **1** | **5** | **-** |
| | **Core Major –II (Theory of Computation)** | **1** | **5** | **-** | **Core Major –II (Network Security and Cryptography)** | **1** | **5** | **-** |
| | **Core Major –III (Advance Operating Systems)** | **1** | **5** | | **Core Major –III (Software Engineering)** | **1** | **5** | **-** |
| | **Core Major Elective – I* (*Machine Learning *Cloud Computing)** | **1** | **4** | **-** | **Core Major Elective – II* (*Python Programming *Image Processing)** | **1** | **4** | **-** |
| | **Major Practical (Shell Programming)** | **1** | **-** | **5** | **Major Practical (Compiler Design Lab)** | **1** | **-** | **5** |
| **Minor (One)** | **Minor–I [#] (Natural Language Processing)** | **1** | **2** | **2** | **Minor–II [#] (Data Minning)** | **1** | **2** | **2** |
| **Total** | | **6** | **21** | **7** | | **6** | **21** | **7** |
| **NHEQF Level-6** | ***Student on exit after successfully completing four years (i.e., securing minimum required 176 credits along with securing additional 2 credits under SSD course work) will be awarded "Four years Bachelor's Degree (Honours)", in related field/discipline*** | | | | | | | |
| | | | | | | | | |
| ***Note: * In case of Core Major Elective course, if the department want to introduce practical component, the department may bifurcate the total 4 credits between theory and practical.*** <br> ***Note: [#] If the minor course is offered without a practical component, the department must allocate 4 credits to the theory component. Electives may be offered by the departments under the Minor.*** | | | | | | | | |
| **Minor–I*** Each department will have to prepare Minor course (One in each semester), which enriches the learner's knowledge beyond the Major discipline (Core Major). The minor courses opted by any learner should be different from the Core Major offered by the Department. | | | | | | | | |

## Fourth Year- (U.G. Honours with Research)

**The following course structure under FYUP is designed for subjects which have practical based courses or have relatively larger emphasis on practical course-based learning.**

# B.Sc. Computer Science Syllabus NEP 2020 (Batches 2022 to 2025)

**(For practical based subjects)**

| **Entry requirement** | (After completing requirements of a 3-year bachelor's degree (120 credits) and 2 additional credits under SSD, candidates who meet a minimum CGPA of 7.5 will be allowed to continue studies in the fourth year of the undergraduate programme leading to the four years bachelor's degree (Honours with Research). | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Course Type** | **Semester-VII** | | | | **Semester-VIII** | | | |
| | **Subject/Title** | **No. of paper** | **Credits** | | **Subject /Title** | **No. of paper** | **Credits** | |
| | | | **T** | **P** | | | **T** | **P** |
| **Core Subject (One)** | **Core Major -I (Computer Organization and Architecture)** | **1** | **5** | **-** | **Core Major-I (Compiler Design)** | **1** | **5** | **-** |
| | **Core Major –II (Theory of Computation)** | **1** | **5** | **-** | | | | |
| | **Core Major Elective-I* (*Machine Learning *Cloud Computing)** | **1** | **4** | **-** | **Core Major Elective -I* (*Python Programming *Image Processing)** | **1** | **4** | **-** |
| | **Major Practical (Digital Electronics Lab)** | **1** | **-** | **5** | **Major Practical (Compiler Design Lab)** | **1** | | **3** |
| | **Research Methodology** | **1** | **5** | | **Dissertation** | **1** | | **12** |
| **Minor (One)** | **Minor–I # (Natural Language Processing)** | **1** | **2** | **2** | **Minor–II # (Data Minning)** | **1** | **2** | **2** |
| **Total** | | **6** | **21** | **7** | | **5** | **11** | **17** |
| **NHEQF Level-6** | ***Student on exit after successfully completing four years (i.e., securing minimum required 176 credits along with securing additional 2 credits under SSD course work) will be awarded "Four years Bachelor's Degree (Honours with Research)", in related field/discipline*** | | | | | | | |

***Note: * In case of Core Major Elective course, if the department want to introduce practical component, the department may bifurcate the total 4 credits between theory and practical.***

***Note: # If the minor course is offered without a practical component, the department must allocate 4 credits to the theory component. Electives may be offered by the departments under the Minor courses.***

**Minor–I*** Each department will have to prepare Minor course (One in each semester), which enriches the learner's knowledge beyond the Major discipline (Core Major). The minor courses opted by any learner should be different from the Core Major offered by the Department.

If a student selects a minor course from a particular subject or department, they are required to study the courses offered by that same subject/department in both the 7th and 8th semesters.

***Important Note:*** *The student may select Minor course either from his/her second core, studied up to 6th semester, or may select from the I.D/M.D subject they have pursued in the first and second year of their U.G. Programme.*

**For Example:** If a student has passed U.G. 3 years with two core subjects i.e. Zoology and Botany, and the student have opted for Zoology as his/her Major subject in the 4th year, then the student may opt Minor courses (To be studied in 7th and 8th semester) from any one subject, which could be either Botany or ID/MD subject studied by him/her in first two years of FYUP.

# B.Sc. Computer Science Syllabus NEP 2020 (Batches 2022 to 2025)

**SOS/CSE/MJ/C701: Computer Organization and Architecture**

**Course Objective:**

This course aims to provide a detailed understanding of the fundamental concepts of computer organization and architecture. Students will explore how computers represent data, perform computations, manage memory, and communicate with peripheral devices. The course includes topics such as logic circuits, hardware structure, instruction execution, input/output operations, and memory management. Through theoretical concepts and practical design exposure, learners will gain insight into the internal workings of modern computing systems.

**Course Content/Syllabus:**

**Unit 1: Representation of Information and Basic Building Blocks**

Number systems: Binary, Octal, Hexadecimal and their conversions; Character codes: BCD, ASCII, EBCDIC; Digital codes: Gray Code, XS-3 Code.

**Unit 2: Logic Circuits**

Basic logic functions; Synthesis of logic using AND, OR, NOT gates; Minimization of logic expressions; Synthesis using NAND and NOR gates; Implementation of logic gates; Flip-Flops, Registers and Shift Registers, Counters, Decoders, Multiplexers, Programmable Logic Devices; Sequential circuits.

**Unit 3: Basic Structure of Computer Hardware and Software**

Functional units of a computer; Basic operational concepts; Bus structures; Introduction to software; Performance evaluation; Concepts of distributed computing.

Basic concepts of memory locations and operations; Addressing modes; Basic I/O operations; Data structures: stacks and queues; Subroutine implementation and use.

**Unit 4: Processing Unit**

Fundamental concepts of instruction execution; Hardwired vs. Microprogrammed control; Performance considerations; Arithmetic operations (signed addition/subtraction, multiplication, division); Branching conditions; Floating-point numbers and arithmetic operations.

**Unit 5: Input-Output Organization**

Accessing I/O devices; Interrupt handling; Direct Memory Access (DMA); I/O hardware; Standard I/O interfaces.

**Unit 6: Memory Organization**

Semiconductor RAM and ROM; Cache memory and performance considerations; Virtual memory concepts; Memory management requirements and techniques.

Introduction to Computer Peripherals Overview of input/output devices; On-line storage concepts and devices.

**References:**


1. Willam Stalling, "Computer Organization and Architecture" Pearson Education Asia
2. Mano Morris, "Computer System Architecture" PHI
3. Zaky and Hamacher, "Computer Organization: McGraw Hill
4. B. Ram, "Computer Fundamental Architecture and Organization" New Age

# B.Sc. Computer Science Syllabus NEP 2020 (Batches 2022 to 2025)

**Course: Theory of Computation (SOS/CSE/MJ/C702)**

**Course Objective:**

This course introduces students to the foundational concepts of computation theory. It covers mathematical models of computation such as finite automata, pushdown automata, and Turing machines. The course emphasizes the classification of formal languages, grammar simplification, and decision algorithms. Through this, students develop a rigorous understanding of what problems can be solved computationally and the limits of computing machines.

**Course Content/Syllabus:**

**Unit 1 - Introduction to the Theory of Computation and Finite Automata:** Mathematical preliminaries and Notation, three basic concepts, applications, deterministic Finite Acceptors, Nondeterministic finite acceptors, equivalence of Deterministic and Nondeterministic finite acceptors, reduction of the Number of states in Finite Automata.

**Unit 2 - Regular Languages, Regular Grammars and Properties of Regular Languages:** Regular expressions, connection between regular expressions and regular languages, regular grammars, closure properties of regular languages, elementary questions about regular languages, identifying language.

**Unit 3 - Context-Free Languages and Simplification of Context-Free Grammars and Normal Forms:** Context-free grammars, parsing and ambiguity, context-free grammars and programming languages, methods of transforming grammars, two important normal forms.

**Unit 4 - Pushdown Automata and Properties of Context-Free Languages:** Nondeterministic pushdown automata, pushdown automata and context-free language, deterministic pushdown automata and deterministic context-free languages, two pumping lemmas, closure properties and decision algorithms for context-free language.

**Unit 5 - Turing Machines and Other Models of Turing Machines:** The standard Turing machine, combining Turing machines for complicated tasks, Turing's thesis, a minor variation on the Turing machine, combining Turing machines, a universal Turing machine.

**Recommended Books:**


1. *An Introduction to Formal Languages and Automata* – Peter Linz, Narosa Publishing House, 1997.
2. *Introduction to Languages and the Theory of Automata* – John C. Martin, McGraw-Hill, 1997.
3. *Introduction to Automata Theory, Languages, and Computation* – J.P. Hopcroft, J.D. Ullman, Narosa Publishing.

# B.Sc. Computer Science Syllabus NEP 2020 (Batches 2022 to 2025)

**Course: Advanced Operating System (SOS/CSE/MJ/C703)**

**Course Objective:**

This course is designed to provide students with an in-depth understanding of the design, implementation, and functioning of advanced operating systems. It covers memory management, CPU scheduling, process synchronization, deadlocks, device and file system management, as well as the structure and characteristics of various types of operating systems such as batch systems, time-sharing systems, distributed systems, and real-time systems. Through this course, students will develop a thorough understanding of how an OS manages system resources and ensures efficient execution of programs.

**Course Content/Syllabus:**

**Unit 1 - Introduction:** What is an Operating System, Simple Batch Systems, Multiprogrammed Batch Systems, Time-Sharing Systems, Personal Computer Systems, Parallel Systems, Distributed Systems, Real-Time Systems.

**Unit 2 - Memory Management:** Background, Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging. Virtual Memory: Demand Paging, Page Replacement, Page Replacement Algorithms, Performance of Demand Paging, Allocation of Frames, Thrashing, Other Considerations, Demand Segmentation.

**Unit 3 - Processes:** Process Concept, Process Scheduling, Operation on Processes, Cooperating Processes, Interprocess Communication. **CPU Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Algorithm Evaluation. **Process Synchronization:** Background, The Critical-Section Problem, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Critical Regions, Monitors, Synchronization in Solaris 2, Atomic Transactions. **Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock, Combined Approach to Deadlock Handling.

**Unit 4 - Device Management:** Techniques for Device Management, Dedicated Devices, Shared Devices, Virtual Devices; Device Characteristics – Hardware Consideration, Input or Output Devices, Storage Devices, Channels and Control Units, Independent Device Operation, Buffering, Multiple Paths, Block Multiplexing, Device Allocation Consideration. **Secondary-Storage Structure:** Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, Disk Reliability, Stable-Storage Implementation.

**Unit 5 - Information Management:** Introduction, A Simple File System, General Model of a File System, Symbolic File System, Basic File System, Access Control Verification, Logical File System, Physical File System. **File-System Interface:** File Concept, Access Methods, Directory Structure, Protection, Consistency Semantics. **File-System Implementation:** File-System Structure, Allocation Methods, Free-Space Management, Directory Implementation, Efficiency and Performance, Recovery.

**Recommended Books:**


1. Abraham Silberschatz and Peter Baer Galvin, *Operating System Concepts*
2. Milan Milankovic, *Operating Systems: Concepts and Design*, McGraw-Hill
3. R. C. Joshi, *Operating System*, Wiley Dreamtech India Pvt. Ltd.
4. Harvey M. Deitel, *Operating Systems*, Addison-Wesley

**SOS/CSE/MJ/E1.1: Machine Learning**

**Course Objective:**

This course aims to introduce students to the foundational concepts and techniques of machine learning. It covers various learning paradigms, including supervised, unsupervised, and reinforcement learning, and dives into core algorithms such as decision trees, Bayesian models, neural networks, genetic algorithms, and clustering techniques. The course is designed to equip students with both theoretical knowledge and practical skills to design, implement, and evaluate machine learning systems for real-world problems.

**Course Content :**

**Unit 1 -** Introduction: Well-Posed learning problems, Basic concepts, Designing a learning system, Issues in machine learning. Types of machine learning: Learning associations, Supervised learning, Unsupervised learning, Reinforcement learning

**Unit 2 -** Decision Tree Learning: Decision tree representation, appropriate problems for decision tree learning, Univariate Trees (Classification and Regression), Multivariate Trees, Basic Decision Tree Learning algorithms, Hypothesis space search in decision tree learning, Inductiv bias in decision tree learning, Issues in decision tree learning.

**Unit 3 -** Bayesian Learning: Bayes theorem and concept learning, Bayes optimal classifier, Gibbs algorithms, Naive Bayes Classifier, Bayesian belief networks, The EM algorithm. Artificial Neural Network: Neural network representation, Neural Networks as a paradigm for parallel processing, Linear discrimination, pairwise separation, Gradient Descent, Logistic discrimination, Perceptron, Training a perceptron, Multilayer perceptron, Back propagation Algorithm. Recurrent Networks, dynamically modifying network structure.

**Unit 4 -** Genetic Algorithms: Basic concepts, Hypothesis space search, Genetic programming, Models of evolution and learning, Parallelizing Genetic Algorithms.

**Unit 5 -** Data Mining Techniques for Analysis: Classification: Decision tree induction, Bayes classification, Rule-based classification, Support Vector Machines, Classification Using Frequent Patterns, k-Nearest-Neighbor, Fuzzy-set approach Classifier, Clustering: K-Means, k-Medoids, Agglomerative versus Divisive Hierarchical Clustering Distance Measures in Algorithmic Methods, Mean-shift Clustering.

**References:**

1. Mitchell T.M., Machine Learning, McGraw Hill
2. Bishop C., Pattern Recognition and Machine Learning, Springer Verlag.
3. Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press

**SOS/CSE/MJ/E1.2 : Cloud Computing**

**Course Objectives**

- To understand the various distributed system models and evolving computing paradigms
- To gain knowledge in virtualization of computer resources.
- To realize the reasons for migrating into cloud.
- To introduce the various levels of services that can be achieved by a cloud.
- To describe the security aspects in cloud and the services offered by a cloud.

**Course Content:**

**UNIT - I**

Cloud Computing Fundamentals: Definition of Cloud computing, Roots of Cloud Computing, Layers and Types of Clouds, Desired Features of a Cloud, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers. Computing Paradigms: High-Performance Computing, Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing.

**UNIT- II**

Migrating into a Cloud: Introduction, Broad Approaches to Migrating into the Cloud, the Seven-Step Model of Migration into a Cloud. Virtualization: Virtual Machines and Virtualization of Clusters and data centers - Implementation Levels of Virtualization -Virtualization Structures/Tools and Mechanisms- Virtualization of CPU, Memory, and I/O Devices-Virtual Clusters and Data Centers.

**UNIT- III**

Infrastructure as a Service (IAAS) & Platform (PAAS): Virtual machines provisioning and Migration services, Virtual Machines Provisioning and Manageability, Virtual Machine Migration Services, VM Provisioning and Migration in Action. On the Management of Virtual machines for Cloud Infrastructures- Aneka—Integration of Private and Public Clouds.

**UNIT- IV**

Software as a Service (SAAS) &Data Security in the Cloud: Software as a Service SAAS), Google App Engine – Centralizing Email Communications- Collaborating via Web Based Communication Tools-An Introduction to the idea of Data Security.

**UNIT- V**

SLA Management in cloud computing: Traditional Approaches to SLO Management, Typesof SLA, Life Cycle of SLA, SLA Management in Cloud.

**References:**

1. Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi, Mastering Cloud Computing: Foundations and Applications Programming, McGraw Hill Education.
2. Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, Cloud Computing: A Practical Approach, McGraw Hill Education.

**SOS/CSE/MI/7.1: Natural Language Processing**

**Course Objective:**
This course aims to introduce students to the fundamental concepts, models, and algorithms used in natural language processing (NLP). The course covers core areas including morphological analysis, syntactic and semantic processing, and language generation. Students will learn how natural language is processed computationally using finite-state automata, grammar rules, probabilistic models, and machine learning techniques. The course also introduces practical applications such as machine translation, information retrieval, and conversational agents.

**Course Content :**

**Unit 1 -** Regular expressions and automata, Morphology and Finite State transducers, N – grams.

**Unit 2 -** Word classes and part of speech tagging, Context free grammars for English, Parsing with context free grammars.

**Unit 3 -** Features and Unifications, Lexicalized and Probabilistic parsing.

**Unit 4 -** Semantics: Representing meaning, Semantic analysis, Lexical semantics, Word Scene Disambiguation and Information retrieval.

**Unit 5 -** Pragmatics: Discourse, Dialog and Conversational Agents, Natural Language Generation, Machine Translation.

**References:**

1. Daniel, Jurafsky and Martin, Speech and Language Processing, Pearson, 2003

# B.Sc. Computer Science Syllabus NEP 2020 (Batches 2022 to 2025)

**SOS/CSE/MJ/801: Compiler Design**

**Course Objectives:**

- To understand the structure and functionality of a compiler.
- To study the various phases involved in the compilation process.
- To learn lexical analysis, syntax analysis, semantic analysis, intermediate code generation, and code optimization techniques.
- To explore the use of tools like LEX and YACC for compiler construction.
- To develop the ability to design and implement a simple compiler.

**Course Content:**

**Unit 1 -** Compiler Structure: Compilers and Translators, Various Phases of Compiler, Pass Structure of Compiler, Bootstrapping of Compiler. Programming Language: High level languages, lexical and syntactic structure of a language, Data elements, Data Structure, Operations, Assignments, Program unit, Data Environments, Parameter Transmission.

**Unit 2 -** Lexical Analysis: The role of Lexical Analyzer, A Simple approach to the design of Lexical Analyzer, Regular Expressions, Transition Diagrams, Finite state Machines, Implementation of Lexical Analyzer, Lexical Analyzer Generator: LEX, Capabilities of Lexical Analyzer.

**Unit 3 -** The Syntactic Specification of Programming Languages: CFG, Derivation and Parse tree, Ambiguity, Capabilities of CFG. Basic Parsing Techniques: Top-Down parsers with backtracking, Recursive descent Parsers, Predictive Parser, Bottom-up Parsers, Shift-Reduce Parsing, Operator Precedence Parsers, LR parsers (SLR, Canonical LR, LALR) Syntax Analyzer Generator: YACC

**Unit 4 -** Intermediate Code Generation: Different Intermediate forms: Three address code, Quadruples and Triples, Syntax Directed Translation mechanism and attributed definition. Translation of Declaration, Assignment, Control flow, Boolean expression, Array References in arithmetic expressions, procedure calls, case statements, postfix translation. Run Time Memory Management: Static and Dynamic storage allocation, stack-based memory allocation schemes, Symbol Table management. Error Detection and Recovery: Lexical phase errors. Syntactic phase errors, semantic errors.

**Unit 5 -** Code Optimization and Code Generation: Local optimization, Peephole optimization, Basic blocks and flow Graphs, DAG, Data flow analyzer, Machine Model, Order of evaluation, Register allocation and code selection.

**References:**


1. Alfred V Aho, Jeffrey D. Ullman, "Principles of Compiler Design", Narosa
2. A.V. Aho, R. Sethi and J.D.Ullman, "Compiler Principle, Tech and tools" AW
3. H.C. Holub "Compiler Design in C", Printice Hall Inc.
4. Apple, "Modern Computer Implementation in C: Basic Design" Cambridge Press
5. Modern Compiler Design: Dick Grune, Wiley dreamtech India Pvt. Ltd.
6. Starting Out with Modern Compiler " David Gaddis Wiley dreamtech India Pvt. Ltd.

**SOS/CSE/MJ/C802: Network Security and Cryptography**

**Course Objectives:**

- To introduce the fundamental concepts of cryptography and network security.
- To explore various encryption techniques, including both classical and modern ciphers.
- To understand public key infrastructure, digital signatures, and key management.
- To study cryptographic hash functions and message authentication codes.
- To understand authentication protocols, web and email security mechanisms.

**Course Content:**

**Unit 1 -** Introduction of Cryptography: Introduction To security: Attacks, Services and Mechanisms, Security, Attacks, Security Services, Conventional Encryption: Classical Techniques, Conventional Encryption Model, and steganography, Classical Encryption Techniques. Modern Techniques: Simplified DES, Block Cipher Principles, DES Standard, DES Strength, Differential and Linear Cryptanalysis, Block Cipher Design Principles, Block Cipher Modes of Operations.

**Unit 2 -** Conventional Encryption Algorithms: Triples DES, Blowfish, International Data Encryption Algorithm, RCS, CAST-128, CR2 Placement and Encryption Function, Key Distribution, Random Number Generation, Placement of Encryption Function.

**Unit 3 -** Public Key Encryption: Public-Key Cryptography: Principles of Public-Key Cryptosystems, RSA Algorithm, Key, Key Management, Fermat's and Euler's Theorm, Primality, Chinese Remainder Theorem. Hash Functions: Message Authentication and Hash Functions: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Function Birthday Attacks, Security of Hash Function and MACS, MD5 Message Digest Algorithm, Secure Hash Algorithm (SHA), Digital Signatures: Digital Signature, Authentication Protocol, Digital Signature Standard (DDS) Proof of Digital Signature Algorithm.

**Unit 4 -** Network and System Security: Authentication Applications: Kerberos X-509, Directory Authentication Service, Electronic Mail Security, Pretty Good Privacy (PGP),S/Mine Security: Architecture, Authentication Header, Encapsulating Security Payloads, Combining Security Associations, Key Management, Web Security: Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction (Set), System Security: Intruders, Viruses, Firewall Design Principles, Trusted Systems.

**References:**

1. William Stallings, "Cryptography and Network Security: Principles and Practice"' Prentice hall, New Jersey
2. Johannes A. Buchmann, "Introduction to Cryptography" Springer-Verlag
3. Atul Kahate, "Cryptography and Network Security" TMH
4. Network Security Bible : Eric Cole, Wiley dreamtech India Pvt. Ltd.
5. 5.Practical Cryptography "Bruce Schneier" Wiley dreamtech India Pvt. Ltd.

**SOS/CSE/MJ/C803: Software Engineering**

**Course Objectives:**

- To understand the fundamental concepts and importance of software engineering in the development of high-quality software.
- To familiarize students with software development life cycle models and requirement engineering.
- To explore design principles, approaches, and coding strategies used in software development.
- To study different software testing strategies and quality assurance techniques.
- To understand project management aspects such as cost estimation, scheduling, risk analysis, and configuration management.
- To introduce the use and architecture of CASE tools and environments.

**Course Content:**

**Unit 1 - Introduction:** Introduction to software engineering, Importance of software, evolving role of software, Software Characteristics, Software Components, Software Applications, Software Crisis, Software engineering problems, Software Development Life Cycle, Software Process. Software Requirement Specification: Analysis, Principles, Water Fall Model, The Incremental Model, Prototyping, Spiral Model, Role of management in software development, Role of matrices and Measurement, Problem Analysis, Requirement specification, Monitoring and Control.

**Unit 2 - Software-Design:** Design principles, problem partitioning, abstraction, top down and bottom up-design, Structured approach functional versus object oriented approach, design specifications and verification, Monitoring and control, Cohesiveness, coupling, Forth generation techniques, Functional independence, Software Architecture, Transaction and Transaction and Transform Mapping, Component level Design, Forth Generation Techniques. Coding: Top-Down and Bottom-Up programming, structured programming, information hiding, programming style and internal documentation.

**Unit 3 - Testing principles:** Levels of testing, functional testing, structural testing, test plane, test case specification, reliability assessment, software testing strategies, Verification and validation, Unit testing, Integration Testing, Alpha and Beta testing, system testing and debugging.

**Unit 4 - Software Project Management:** The Management spectrum (The people, the product, the process, the project) Cost estimation, project scheduling, staffing, software configuration management, Structured Vs. Unstructured maintenance, quality assurance, project monitoring, risk management. Software Reliability and Quality Assurance: Reliability issues, Reliability metrics, Reliability growth modeling, Software quality, ISO 9000 Certification for software industry, SEI capability maturity model, comparison between ISO and SEI CMM.

**Unit 5 - CASE (Computer Aided Software Engineering):** CASE and its scope, CASE support in software life cycle, documentation, project management, internal interface, Reverse Software Engineering, Architecture of CASE environment.

**References:**

1. Pressman, Roger S., "Software Engineering: A Practitioner's Approach Ed. oston: McGraw Hill
2. Jalote, Pankaj, "Software Engineering Ed.2"New Delhi: Narosa 2002
3. Schaum's Series, "Software Engineering" TMH
4. Ghezzi Carlo and Others "Fundamentals of Software Engineering" PHI
5. Alexis, Leon and Mathews Leon, "Fundamental of Software Engg.

# B.Sc. Computer Science Syllabus NEP 2020 (Batches 2022 to 2025)

**SOS/CSE/MJ/E2.1: Python Programming**

**Course Objectives:**

- To understand the fundamental concepts and importance of software engineering in the development of high-quality software.
- To familiarize students with software development life cycle models and requirement engineering.
- To explore design principles, approaches, and coding strategies used in software development.
- To study different software testing strategies and quality assurance techniques.
- To understand project management aspects such as cost estimation, scheduling, risk analysis, and configuration management.
- To introduce the use and architecture of CASE tools and environments.

**Course Content**

**Unit-1** Introduction to Programming

Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs
in Python.

**Unit-2** Creating Python Programs

Identifiers and keywords; Literals, numbers, and strings; Operators; Expressions; Input/output statements;
Defining functions; Control structures (conditional statements, loop control statements, break, continue and
pass, exit function), default arguments.

**Unit-3** Built-in data structures

Mutable and immutable objects; Strings, built-in functions for string, string traversal, string operators and
operations; Lists creation, traversal, slicing and splitting operations, passing list to a function; Tuples, sets,
dictionaries and their operations.

**Unit 4** Object Oriented Programming

Introduction to classes, objects and methods; Standard libraries

**Unit 5** File and exception handling

File handling through libraries; Errors and exception handling.

**References:**


1. Python Programming – S. Taneja
2. Introduction to Computing and Problem-Solving using Python – E. Balaguruswamy

**SOS/CSE/MJ/E2.2 : Image Processing**

**Course Objectives**

- To introduce the fundamental concepts of digital image processing and its applications.
- To understand the basic components of an image processing system and visual perception.
- To study techniques for image enhancement in both spatial and frequency domains.
- To analyze image restoration models and compression techniques for efficient image storage and transmission.
- To understand morphological operations and segmentation techniques used for object recognition and analysis.
- To develop skills for implementing image processing algorithms for practical applications.

**Course Content:**

**Unit 1** Introduction: Digital Image Processing, The origins of Digital Image Processing, Examples of Digital Image Processing application, Fundamental steps in Digital Image processing, Components of Image Processing system Fundamentals: Elements of Visual Perception, Light and Electromagnetic Spectrum, Image Sensing and Acquisition, Image Sampling and Quantization, Some basic Relationships between Pixels, Linear and

**Unit 2** Nonlinear Operations. Image Enhancement in the spatial domain: Background, Some basic gray level transformation, Introduction of Histogram processing, Enhancement using Arithmetic/Logic operations, Basics of spatial filtering, Smoothing spatial filters, Sharpening spatial filters, Image Enhancement in the Frequency Domain : Introduction.

**Unit 3** Image Restoration: Model of the Image Degradation/Restoration process, Noise Models, Restoration in the presence of noise only spatial filtering, Inverse filtering, Minimum Mean Square Error (Wiener) filtering, Geometric mean filter, Geometric Transformations, Image Compression: Fundamentals, Lossy Compression, Lossless Compression, Image Compression models, Error-free Compression : Variable length coding, LZW coding, Bit plane coding, Run length coding, Introduction to JPEG.

**Unit 4** Morphology: Dilation, Erosion, Opening and Closing, Hit-and Miss transform, Morphological Algorithms : Boundry Extraction, Region filling, Extraction of connected components, Convex Hull, Image Segmentation: Definition, characteristics of segmentation Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region based segmentation. Introduction to Representation and Description, Introduction to Object Recognition. References: 1. Digital Image Processing: Rafael C. Gonzalez and Richard E.Woods. Addision Wesley. 2. Fundamentals of Digital Image Processing. Anil K. Jain, PHI. 3. Digital Image Processing and Analysis : B. Chanda and D. Dutta Majumber, PHI. 4. Image Processing in C : Dwayne Phillips, BPB.

**References:**

1. Digital Image Processing: Rafael C. Gonzalez and Richard E.Woods. Addision Wesley.
2. Fundamentals of Digital Image Processing. Anil K. Jain, PHI.
3. Digital Image Processing and Analysis : B. Chanda and D. Dutta Majumber, PHI.
4. Image Processing in C : Dwayne Phillips, BPB.

**SOS/CSE/MI/8.1: Data Mining**

**Course Objectives**

- To introduce the basic concepts and necessity of data mining and data warehousing.
- To understand the architecture and components of data warehouses and OLAP systems.
- To explore data mining primitives and techniques for preprocessing, integration, and transformation.
- To learn various algorithms and methods for association rule mining, classification, and prediction.
- To study clustering techniques and their scalability for large datasets.

**Course Content:**

**Unit 1 -** Introduction to data mining, need for data warehousing and data mining, application potential, keywords and techniques. Data Warehousing and Online analytical Processing (OLAP): Aggregation operations, models for data warehousing, star schema, fact and dimension tables , conceptualization of data warehouse and multidimensional databases, Relationship between warehouse and mining.

**Unit 2 -** Data mining primitives: Data preprocessing, data integration, data transformation. Definition and specification of a generic data mining task. Description of Data mining query language with examples. Association analysis: Different methods for mining association rules in transaction-based data bases. Illustration of confidence and support. Multidimensional and multilevel association rules. Classification of association rules. Association rule algorithms – A priori and frequent pattern growth.

**Unit 3 -** Classification and Prediction: Different classification algorithms. Use of genie index, decision tree induction, Bayesian classification, neural network technique of back propagation, fuzzy set theory and genetic algorithms. Clustering: Partition based clustering, hierarchical clustering, model-based clustering for continuous and discrete data. Scalability of clustering algorithms. Parallel approaches for clustering.

**Unit 4 -** Web mining: Web usage mining, web content mining, web log attributes. Data mining issues in object-oriented data bases, spatial data bases and multimedia data bases and text data bases.

**References:**

1. J. Han, M. Kamber, “Data Mining Concepts and Techniques”, Harcourt India Pvt Ltd, 2001
2. M. Dunham, “ Data Mining : introductory and Advanced Topics”, Pearson Pub, 2003
3. A.K. Pujari, “Data Mining Techniques”, Universities Press.

**SOS/CSE/C404: Research Methodology**

**Course Objective:**

To equip students with the foundational understanding of research methodologies, including the identification of research problems, formulation of research design, and application of qualitative and quantitative approaches, along with sampling and data analysis techniques.

**Course Content:**

**Unit 1 -** Meaning of Research: Function of Research, Research Characteristics, Steps involved in Research, Significance of Research, Types of Research, Criteria of Good Research, Research in Pure and Applied Sciences, Areas of Science, Philosophy of Science, Interdisciplinary Research, Review of Literature.

**Unit 2 -** Identification of Research Problem: Selecting the Research Problem, Defining the Problem, Goals and Criteria for Identifying problems for research, Techniques Involved in Defining the Problem, Source of Problems, Personal Consideration.

**Unit 3 -** Research Design: Concept and Importance in Research, Need for Research Design, Formulation of Research Design, Exploratory Research Design, Descriptive Research Designs, Experimental Design, Basic principles of experimental designs, Computer and Internet in designs.

**Unit 4 -** Qualitative and Quantitative Research: Qualitative research, Quantitative research, Concept of measurement, causality, generalization, replication, Merging the two approaches. Sampling and Data Analysis: Sampling Frame, Sample Size, Characteristics of a Good Sample, Random Sampling, Systematic Sampling, Practical Considerations in Sampling and Sample Size, Data Preparation, Univariate and Bivariate analyses.

**Reference Books:**

1. Research Methodology: Methods & Techniques, C.R.Kothari, New Age International Publishers.
2. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, John W. Creswell, SAGE Publications Inc.
3. Research Methodology, Mukul Gupta & Deepa Gupta, PHI Learning Private Ltd, New Delhi.
4. Research Methodology, Uma Sekaran & Roger Bougie, NMIMS Global Access
5. The Scientific Endeavor, Jeffrey A. Lee, Pearson India