SIES (Nerul) College of Arts, Science and Commerce (Autonomous)

(Affiliated to University of Mumbai)
RE-ACCREDITED GRADE "A" BY NAAC (3rd CYCLE)

BOARD OF STUDIES
SYLLABUS FOR
M.Sc. (Computer Science) Part 1

(WITH EFFECT FROM THE ACADEMIC YEAR 2026-2027)

OBJECTIVES OF THE PROGRAMME

1. To be fundamentally strong at core subject of Computer Science.
2. To apply programming and computational skills for industrial solutions.
3. Broad understanding of latest technological trends.
4. To identify opportunities for establishing an enterprise for immediate Employment.
5. Able to understand and apply fundamental research concepts.

PROGRAMME OUTCOMES:

1. An ability to apply the skills acquired in post-graduation to get better career prospects.
2. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computational systems.
3. An ability to apply the knowledge for higher research in a specialized area of Computer Science.
4. Ability to understand and apply the core concepts of Computer Science.
5. Ability to provide socially acceptable technical solutions in various domains of specializations.

SIES(Nerul) College of Arts, Science and Commerce (Autonomous)
NEP VERTICLES FOR
M.Sc. (Computer Science) Part 1

Semester	Major	Major Elective (Any One) NPTEL	Minor / RM	OJT, FP, CEP, CC, RP	Cum Cr/ Sem.
I	Image Processing (3+1P Credits) Algorithm for Optimization (3+1P Credits) Advanced Database Techniques (3+1P Credit) Mini Project in Emerging Technologies (2 Credits)	Software Defined Networking (3+1P Credits) (OR) Cloud Computing Devops (3+1P credits)	Research Information Systems and Computing (4 Credit)		-
Total	**14**	**4**	**4**	0	**22**

Image Processing

COURSE CODE: **COURSE CREDIT: 03**

1 credit - 15 lectures

1 lecture - 60 minutes

Course Objectives:

- To introduce students to the principles of Digital Image Processing, including its applications, image formation, visual perception, sampling, quantization, pixel relationships, and the basic mathematical tools used in image processing.
- To familiarize students with image enhancement techniques using intensity transformations, spatial filtering, and frequency-domain methods based on Fourier transform concepts.
- To Develop skills to design and implement algorithms for advanced image analysis.

Course Outcomes:

After successful completion of the course, the student will be able to:

- Explain and analyze digital images by applying concepts of image acquisition, sampling, quantization, pixel relationships, and fundamental mathematical tools used in Digital Image Processing.
- Apply intensity transformation, spatial filtering, and frequency-domain filtering techniques to enhance and analyze digital images.
- Knowledge of Perform frequency domain operations on images.

Unit	Syllabus	No. of lectures
1.	Introduction: What is Digital Image Processing? The Origins of Digital Image Processing, Examples of Fields that Use Digital Image Processing, Fundamental Steps in Digital Image Processing, Components of an Image Processing System Digital Image Fundamentals: Elements of Visual Perception, Light and the Electromagnetic Spectrum, Image Sensing and Acquisition, Image Sampling and Quantization, Some Basic Relationships Between Pixels, Introduction to the Basic Mathematical Tools Used in Digital Image Processing	15
	Intensity Transformations and Spatial Filtering: Background, Some Basic Intensity Transformation Functions, Histogram Processing, Fundamentals of Spatial Filtering, Smoothing (Lowpass) Spatial Filters, Sharpening (Highpass) Spatial Filters, Highpass, Bandreject, and Bandpass Filters from Lowpass Filters, Combining Spatial Enhancement Methods	

2.	Filtering in the Frequency Domain: Background, Preliminary Concepts, Sampling and the Fourier Transform of Sampled Functions, The Discrete Fourier Transform of One Variable	**15**
3.	Image Restoration and Reconstruction: A Model of the Image Degradation/Restoration Process, Noise Models, Restoration in the Presence of Noise Only - Spatial Filtering, Periodic Noise Reduction Using Frequency Domain Filtering Color Image Processing: Color Fundamentals, Color Models Case Study	**15**

References:

1. Digital Image Processing by Rafael Gonzalez & Richard Woods, Pearson; 4th edition, 2018
2. Think DSP: Digital Signal Processing in Python by Allen Downey, O'Reilly Media; 1st edition 2014

Additional References:

1. Understanding Digital Image Processing, Vipin Tyagi, CRC Press, 2018
2. Digital Signal and Image Processing by Tamal Bose, John Wiley 2010
3. Hands-On Image Processing with Python by Sandipan Dey,Packt Publishing, 2018
4. Fundamentals of Digital Images Processing by A K Jain, Pearson, 2010

Image Processing - Practical

COURSE CODE : **COURSE CREDIT: 01**

1 credit - 30 lectures

1 lecture is 60 minutes

Sr. No.	List of Practical
1	Write program to demonstrate the Upsampling and downsampling on Image.
2	Write program to demonstrate the Fast Fourier Transform to compute DFT
3	Write program to demonstrate the following aspects on image 1. Convolution operation 2. Template Matching
4	Write program to implement point/pixel intensity transformations such as 1. Log and Power-law transformations 2. Contrast adjustments
5	Write program to implement point/pixel intensity transformations such as 1. Histogram equalization 2. Thresholding, and halftoning operations
6	Write a program to apply various enhancements on images using image derivatives by implementing Gradient and Laplacian operations.
7	Write a program to implement linear and nonlinear noise smoothing on suitable image or sound signal.
8	Write a program to apply various image enhancement using image derivatives by implementing smoothing, sharpening, and unsharp masking filters for generating suitable images for specific application requirements.
9	Write a program to Apply edge detection techniques such as Sobel and Canny to extract meaningful information from the given image samples
10	Write the program to implement various morphological image processing techniques.

MAJOR-Algorithm for Optimization

COURSE CODE : **COURSE CREDIT: 03**

1 credit - 15 lectures

1 lecture is 60 minutes

Course Objectives:

- You will be able to effectively implement optimization techniques to the existing algorithm to improve its performance.
- You will be able to work in the areas of Machine Learning and Data Sciences Algorithms
- You will be able to perform sampling using different algorithms.
- You will be able to deal with Uncertainty in optimization.

Course Outcomes:

- Ability to understand the Optimization Process
- Optimization with a focus on practical algorithms for the design of engineering systems.
- Exposure to multivariable calculus, linear algebra, and probability concepts.
- Learn a wide variety of optimization topics, introducing the underlying mathematical problem formulations and the algorithms for solving them

Sr. No	Syllabus	No. of lectures
01	Unit 01: Introduction to Optimization Process Basic Optimization Problem, Constraints, Critical Points, Conditions for Local Minima, Contour Plots, Unimodality, Fibonacci Search, Golden Section Search, Quadratic Fit Search.Case Study	15
02	Unit 02: Order Methods First-Order Methods, Gradient Descent, Conjugate Gradient, Adagrad, RMSProp, Adadelta, Adam, Hypergradient Descent.Second-Order Methods, Newton's Method, Secant Method, Quasi-Newton Methods.Case Study	15
03	Unit 03: Sampling and Surrogate Models Sampling Plans, Full Factorial, Random Sampling, Uniform Projection Plans, Stratified Sampling, Space-Filling Metrics. Surrogate Models, Fitting Surrogate Models, Linear Models, Fitting Noisy Objective Functions, Model Selection.Bayesian Monte Carlo, Ant Colony Optimization. Case Study	15

References:

1. Think Julia: How to Think Like a Computer Scientist by Allen B. Downey and Ben Lauwens,

1st Edition, 2019, O'Reilly.

2. Decision Making Under Uncertainty: Theory and Application by Mykel J. Kochenderfer,

MIT Lincoln Laboratory Series, 2015.

3. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, 3rd Edition (International Edition), MIT Press, 2009.

MAJOR- Practical Algorithm for Optimization

COURSE CODE: **COURSE CREDIT: 01**

1 credit - 30 lectures

1 lecture is 60 minutes

Note: All the Practical's should be implemented using Julia Link: Julia:https://julialang.org/

Sr. No	List of practical
1	Implement Contour Plots.
2	Implement Fibonacci search
3	Implement Golden section search
4	Implement Quadratic Fit Search.
5	Implement Gradient descent.
6	Implement quasi-Newton methods to find the local maxima.
7	Implement the Adagrad method with application
8	Implement the RMSprop and Adadelta
9	Implement Conjugant Gradient
10	Implement radial basis functions using surrogate modeling.
11	Apply Random Forest in surrogate Model.
12	Implement Gaussian Process and its application.
13	Path finding using Ant Colony Optimization with an application.

MAJOR- Advanced Database Techniques

COURSE CODE: **COURSE**
CREDIT: 03

1 credit - 15 lectures

1 lecture - 60 minutes

Course Objectives:

- To form professional competencies related to design and implementation of non-relational databases, including object-oriented, parallel and Distributed.
- Learners will be able to explore XML, and Mobile databases.
- Learners will be able to deal with methods used for dealing with spatial and Temporal Databases.

Course Outcomes:

- To cover advanced topics of databases to become more proficient.
- To provide students with theoretical knowledge and practical skills in advanced topics in database systems, big data and modern data-intensive systems.
- To Expand Students, view and introduce advanced topics and Business Intelligence.

Unit	Syllabus	No. of lectures
01	**Spatial and Temporal Databases:** Temporal Databases: Overview of Temporal Databases- TSQL2, Spatial Databases- Spatial Data Types- Spatial Relationships- Spatial Data Structures- Spatial Access Methods- Spatial DB Implementation **Distributed Databases**: Concepts, Data fragmentation, Replication and allocation techniques for distributed database design, Query processing, Concurrency control and recovery in distributed databases, Architecture and Design: Centralised versus non centralized Databases, Homogeneous and Heterogeneous DDBMS, Functions and Architecture, Distributed database design, query processing in DDBMS. **NoSQL**: Characteristics of NoSQL, NoSQL Storage types, Advantages and Drawbacks, NoSQL Products Interfacing and interacting with NoSQL: Storing Data In and Accessing Data from MongoDB, Redis, HBase and Apache Cassandra, Language Bindings for NoSQL Data Stores Understanding the storage architecture: Working with Column Oriented Databases, HBase Distributed Storage Architecture, Document Store Internals.	15
02	**Data Warehousing and OLAP** – Introduction to Data Warehousing and its characteristics, objectives and benefits; Need and Applications of Data Warehouse in decision support systems and business intelligence; Difference between OLTP and OLAP with respect to data usage,	

	operations and performance; Data Warehouse Architecture with emphasis on Three-Tier Architecture; Components of Data Warehouse including data sources, ETL tools, storage and metadata; Data Marts – concept, features and types; ETL Process covering Extract, Transform and Load operations; Introduction to OLAP and multidimensional data model; OLAP Operations such as Roll-up, Drill-down, Slice, Dice and Pivot; Simple Case Study on Sales Data Warehouse.	15
03	**Database Security, Backup and Recovery** – Introduction to Database Security and its importance; Need for Database Security in modern database systems; Types of Database Threats including unauthorized access and data leakage; Authentication and Authorization mechanisms; Users, Roles and Privileges management; SQL Injection – Concept, causes and prevention techniques; Database Backup concepts and importance; Types of Backup including Full, Incremental and Differential backups; Database Recovery concepts and failure types; Log-based Recovery and Checkpoint Mechanism; Case Study on Database Failure and Recovery.	15

Textbook(s):

1. Database Management Systems by Raghu Ramakrishnan and Johannes Gehrke, McGraw Hill, 3rd Edition, 2014
2. Database System concepts by Korth
3. Professional NoSQL By Shashank Tiwari, Wrox-John Wiley & Sons, Inc, 2011
4. Getting Started with NoSQL, Gaurav Vaish, Packt Publishing Ltd, 2013

Reference(s):

1. Advanced Database Management System by Rini Chakrabarti and Shilbhadra Dasgupta, Dreamtech Press, 2017
2. SQL & NoSQL Databases, Andreas Meier · Michael Kaufmann, Springer Vieweg, 2019
3. Parallel and Distributed Systems by Arun Kulkarni, Nupur Prasad Giri, Wiley, Second edition, 2017
4. Practical Hadoop Migration: How to Integrate Your RDBMS with the Hadoop Ecosystem and Re-Architect Relational Applications to NoSQL By Bhushan Lakhe, Apress; 1st edition, 2016.

MAJOR- Advanced Database Techniques Practical

COURSE CODE: **COURSE CREDIT: 01**

1 credit - 2 lectures

1 lecture is 60 minutes

Note: All the Practical's should be implemented using NoSQL Link: https://www.oracle.com/database/technologies/nosql-databaseserver-downloads.html.

Sr. No	List of Practical
1	Create different types that include attributes and methods. Define tables for these types by adding a sufficient number of tuples. Demonstrate insert, update and delete operations on these tables. Execute queries on them.
2	Create an XML database and demonstrate insert, update and delete operations on these tables. Issue queries on it.
3	Demonstrate distributed databases environment by dividing given global conceCreate a table that stores spatial data and issue queries on it.
4	Create a table that stores spatial data and issues queries on it.
5	Create a temporal database and issue queries on it.
6	Demonstrate the Accessing and Storing and performing CRUD operations in 21 1. MongoDB 2. Redis
7	Demonstrate the Accessing and Storing and performing CRUD operations in 1. HBase 2. Apache Cassandra
8	Demonstrating MapReduce in MongoDB to count the number of female (F) and male (M) respondents in the database.
9	Perform OLAP operations such as Roll-up, Drill-down, Slice and Dice.
10	Create and analyze a departmental Data Mart.
11	Implement database security using users, roles and privileges.
12	Perform database backup and restore operations
13	Demonstrate database recovery using logs and checkpoint mechanism.

Field Work- Project Work in Emerging Technologies

COURSE CODE: **COURSE CREDIT: 02**

1 credit - 30 lectures

1 lecture - 60 minutes

Course Objectives:

The Project Work aims to:

- Enable students to apply advanced concepts of Computer Science—including AI, ML, Cloud, IoT, Databases, and Mobile Development—to a practical real-life or research-style problem.
- Provide hands-on experience in planning, designing, implementing, testing, and presenting a functional prototype.
- Develop teamwork, communication, ethical responsibility, and technical documentation skills required in modern IT environments.

Course Outcomes:

After successful completion of the course, the student will be able to:

- Design and implement a small-scale but functional solution or research prototype using appropriate technologies and methodologies.
- Prepare clear technical documentation and present the project professionally with a conceptual understanding of design choices and ethical considerations.

Project Types:

A. Real-Life Application Development

Projects solving a practical campus/industry/community problem.

Examples:

- Attendance Monitoring System
- Inventory or Resource Tracking System
- Cybersecurity Log Monitoring Tool
- Academic Software e.g. LMS

B. Innovative Systems or Mini Products

Focused on creativity, prototyping, or system simulation.

Examples:

- Intelligent Reminder/Alert System
- Simple Recommendation Engine
- Lightweight Document Classifier

C. IoT-Based Projects

Projects combining sensors, microcontrollers, and cloud services.

Examples:

- IoT-based Smart Energy Meter
- Home/Office Environmental Monitoring
- IoT Health Alert System (Pulse/Temperature Monitoring)
- Smart Irrigation Controller

D. Mobile Application Development Projects

Android/iOS apps for utility, automation, or productivity.

Examples:

- Mobile App for Event/Attendance Management
- Fitness Tracking App
- Location-based Alert/Reminder App
- Expense Tracker with Local or Cloud Database

E. Research-Oriented Mini Projects

Lightweight research/problem exploration requiring review & experimentation.

Examples:

- Comparative Study of Image Filters
- Performance Analysis of Optimization Algorithms
- Small Dataset ML Evaluation (Classification/Clustering)

Tools & Technologies:

Students are free to choose technologies that suit their problem. Common options include:

- **Programming:** Python, Java, Kotlin, C++, JavaScript
- **AI/ML:** Scikit-Learn, TensorFlow Lite, PyTorch
- **Mobile:** Android Studio (Java/Kotlin), Flutter
- **IoT:** Arduino, ESP32, Raspberry Pi
- **Databases:** MySQL, MongoDB, Firebase, SQLite
- **Cloud:** AWS, GCP, Azure, Thingspeak, Blynk
- **Version Control:** Git/GitHub

. Project Guide:

Assigning a project guide to each project or group is a mandatory requirement to ensure the successful completion of the project work. The guide plays a crucial role as a mentor and technical expert, providing in valuable support and guidance to students. They are expected to facilitate effective communication and teamwork, review project proposals, assign schedules, and monitor progress on a regular basis. Additionally, guides are expected to offer timely feedback, provide guidance on project planning and implementation strategies, evaluate the quality of work, and promote professionalism

and ethical conduct. Their expertise and involvement are essential in helping students navigate challenges, make informed decisions, and achieve their project goals effectively.

Project Team Size: Individual

Project Proposal: The project proposal is a mandatory document that serves as a foundation for the project. It helps students define their project idea, receive early evaluation and feedback, establish clear communication with the project guide, and take ownership of the project's successful execution. A formal proposal ensures systematic and professional project planning, fostering critical thinking, effective communication, and project management skills. The proposal provides a roadmap and increases the chances of a successful outcome. Before initiating a project, it is mandatory to submit a project proposal for approval. **The original duly approved project proposal should be attached to the final project report.** The project proposal for UG computer science projects should include the following contents:

- Title
- Introduction
- Objectives
- Scope
- Methodology
- Tools and Technologies
- Timeline
- Resources
- Expected Outcomes
- References

Project Report:

The Certified Copy of Hard Bound Project Report must adhere to the following guidelines:

- No of Copies: Individual + 1 (College / Department Copy)
- The project report should include the following
    - Title Page *(Sample attached in Appendix)*
    - Certificate *(Sample attached in Appendix)*
    - Declaration *(Sample attached in Appendix)*
    - Acknowledgement
    - Table of Contents
    - Original Copy of approved Project Proposal
    - Self-attested copy of Plagiarism Report from any open source tool.
    - Chapters / Sections depending upon the type of project
    - List of Tables and/or List of Figures
    - References (IEEE / Springer format)
    - Glossary
    - Appendices (Survey datasheets / Questionnaires, ect)
- Use of LaTeX for documentation purposes should be preferred.
- The text of the report should be set in 12 pt, Times New Roman font, and single-spaced.
- Chapter headings should be centered, written in 20 pt, Times New Roman font, bold, and in all caps.
- These guidelines ensure a standardized format for the project report, promoting clarity and readability.

Evaluation:

The evaluation of the project will include a viva voce, which will assess the project based on the following parameters:

- **Documentation – 10 Marks:** The completeness, accuracy, and professionalism of the project documentation, including the project report and supporting materials, will be considered.
- **Quality of the Project – 10 Marks:** The overall quality of the project, including its design, implementation, and user experience, will be evaluated.
- **Working of the Project – 20 Marks**: The functionality and performance of the project will be assessed to determine how well it meets the specified requirements and objectives.
- **Project & Viva Presentation – 10 Marks:** The clarity, organization, and effectiveness of the project presentation will be evaluated. Viva :The viva voce session will provide an opportunity for the student to demonstrate their knowledge and understanding of the project, as well as to answer questions and engage in a discussion with the evaluators.

SAMPLE TITLE PAGE FORMAT

A PROJECT REPORT

on

<PROJECT NAME>

Submitted by

Mr. XYZ

in partial fulfillment for the award of the degree

of

B.Sc. (Honours) Computer Science

under the guidance of

<Guide Name>

Department of Computer Science

<<College Logo>>

<<College Name>>

(Sem VII / VIII)

(202- – 202-)

SAMPLE CERTIFICATE FORMAT

<<College Logo>>

<<College Name>>,

<<College Address>>

Department of Computer Science

CERTIFICATE

This is to certify that Mr./Ms. _____ of **B.Sc. (Honours) Computer Science** class has satisfactorily completed the Project _____, to be submitted in the partial fulfillment for the award of **B.Sc (Honours)Computer Science** during the academic year **202- – 202-**.

Date of Submission:

Project Guide **Head / Incharge,**

Department Computer Science

College Seal **Signature of Examiner**

SAMPLE DECLARATION FORMAT

DECLARATION

I, _____, hereby declare that the project entitled "_____" submitted in the partial fulfilment for the award of **B.Sc (Honours)Computer Science** during the academic year **202- – 202-** is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Signature of the Student:

Place:

Date:

Major Elective- Cloud Computing and DevOps

COURSE CODE: **COURSE CREDIT: 03**

1 credit - 15 lectures

1 lecture - 60 minutes

Course Objectives

- Understand cloud computing concepts, service models, and deployment models
- Use cloud platforms to deploy and manage applications
- Apply DevOps principles for CI/CD automation
- Work with containers and orchestration tools

Course Outcomes

- Explain cloud computing concepts and architectures
- Deploy applications using cloud service models
- Apply DevOps lifecycle and automation principles
- Use containerization and orchestration tools

Unit	Topics	Number of Lectures
I	Introduction to Cloud Computing, Characteristics and benefits of Cloud Computing, Basic concepts of Distributed Systems, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing. Elements of Parallel Computing. Elements of Distributed Computing. Technologies for Distributed Computing. Cloud Computing Architecture. The cloud reference model. Infrastructure as a service. Platform as a service. Software as a service. Types of clouds.	15
II	**Virtualisation:** Characteristics of Virtualised Environments. Taxonomy of Virtualisation Techniques. Virtualisation and Cloud Computing. Pros and Cons of Virtualisation. Virtualisation using KVM, Creating virtual machines, Open challenges of Cloud Computing. **Overview of Cloud Providers:** AWS, Microsoft Azure, Google Cloud Platform. **Managing the data in the Cloud:** Securing data in the cloud, ACL, OAuth, OpenID, XACML, securing data for transport in the cloud, scalability of applications and cloud services.	15
III	**DevOps and Containers in Cloud:** Basics of DevOps: Introduction to DevOps, Continuous Deployment: Containerization Concepts, Docker Architecture, Docker Images & Containers, Dockerfile and Docker, Compose, Introduction to Kubernetes, Kubernetes Architecture & Components, Basic Kubernetes Objects (Pod, Service, Deployment), CI/CD Pipelines using Jenkins / GitHub Actions, Automating Infrastructure on Cloud, Application Deployment and Orchestration using ECS, ECR & EKS	15

TEXTBOOK

1. Kevin L. Jackson. Scott Goessling, Architecting Cloud Computing Solutions,Packt Publishing 2018
2. Shailendra Singh, Cloud Computing: Focuses on the Latest Developments in Cloud Computing, Oxford University Press; First edition, June 2018

REFERENCES

1. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw-Hill Education Private Limited, 2013.

Cloud Computing and DevOps - Practical

COURSE CODE : **COURSE CREDIT: 01**

1 credit - 30 lectures

1 lecture is 60 minutes

Sr. No	List of Practical
1.	Study and implementation of Infrastructure as a Service
2.	Installation and Configuration of Virtualisation Using KVM
3.	Study and implementation of Storage as a Service
4.	Study and implementation of identity management
5.	Study Cloud Security management
6.	Write a program for a web feed.
7.	Study and implementation of Single-Sign-On.
8.	User Management in Cloud.
9.	Git and GitHub Operations: Repository creation, commit, push, pull, and branch operations
10.	Creating an application in Dropbox to store data securely. Develop a source code using the Dropbox API for updating and retrieving files.
11.	Installing and Configuring Docker in localhost and running multiple images on a Docker Platform
12.	Developing an ASP.NET-based web application on the Azure platform
13.	Creating and running Web applications on localhost and deploying the same in Google App Engine

Major Elective - Software Defined Networking

COURSE CODE: **COURSE CREDIT: 03**

1 credit - 15 lectures

1 lecture - 60 minutes

Course Outcome:

- Understand computer networking concepts, OSI/TCP-IP models, and routing protocols.
- Gain knowledge and skills in Software Defined Networking (SDN) architecture, OpenFlow, and application development.
- Comprehend Network Functions Virtualization (NFV), cloud computing, and IoT integration in modern network architectures.
- Design and implement switching techniques, routing protocols, multicast, MPLS, traffic filtering, and routing redistribution.
- Develop network design and deployment skills for efficient and secure routing, traffic management, and integration of network components.

Course Objectives:

- To make the students capable of understanding computer network basics.
- To Obtain the knowledge of Software defined networks with understanding of data plane, control plane and application plane.
- To apply network virtualization for industry standard solutions. To improve skills in implementing network virtualization and Software Defined Network (SDN).

Unit No.	Syllabus	No. of lectures
01	**Module -1- Introduction to Software Defined Networking:** Understanding the layered architecture of OSI/RM and TCP-IP Model, Study of various network Routing protocols, Introduction to Transport layer and Application layer protocols. Elements of Modern Networking, SDN: Background and Motivation, SDN Data Plane and OpenFlow, SDN Control Plane, SDN Application Plane.	15
02	**Module-2 -Network Functions Virtualization:** Concepts and Architecture, NFV Functionality, Network Virtualization Quality of Service, Modern Network Architecture: Cloud and Fog, Cloud Computing, The Internet of Things: Components	15
03	**Module-3 -Design and implementation of Network:** Understand and implement Layer 2/3 switching techniques (VLAN/TRUNKING/Managing Spanning Tree), Implementation of OSPF V2 and V3, Implementation BGP, Implementation Multicast Routing, Implementation of MPLS, Implementation of Traffic Filtering by using Standard and Extended Access Control List, Implementation of Routing redistribution, Implementation of Policy Based Routing/Load Balancing /QOS/Natting /VRF	15

References:

1. Behrouz A Forouzan —TCPIP Protocol Suite‖ Fourth Edition 2010
2. William Stallings, —Foundations of Modern Networking‖, Pearson Ltd.,2016.
3. Software Defined Networks: A Comprehensive Approach by Paul Goransson and Chuck Black, Morgan Kaufmann Publications, 2014
4. SDN - Software Defined Networks by Thomas D. Nadeau & Ken Gray, O'Reilly, 2013
5. Network Programmability and Automation-Jason Edelman, Matt Oswalt First Edition 2018.

Software Defined Networking Practical

COURSE CODE: **COURSE CREDIT: 01**

1 credit - 30 lectures

1 lecture is 60 minute

Note:

All the Practical's should be implemented using GNS3/EVENG/CISCO VIRL

Link: GNS3 :https://www.gns3.com/software/download

EVE-NG: https://www.eve-ng.net/index.php/download/CISCO

VIRL:https://learningnetwork.cisco.com/s/question/0D53i00000Kswpr/virl-15-download

Sr. No	Topics
1	1. Implement SPAN Technologies (Switch Port Analyzer) 2. Implement SNMP and Syslog 3. Implement Flexible NetFlow
2	1. Implement a GRE Tunnel 2. Implement VTP 3. Implement NAT
3	Implement Inter-VLAN Routing
4	Observe STP Topology Changes and Implement RSTP 1. Implement Advanced STP Modifications and Mechanisms 2. Implement MST
5	1. Implement EtherChannel 2. Tune and Optimize EtherChannel Operations
6	OSPF Implementation 1. Implement Single-Area OSPFv2 2. Implement Multi-Area OSPFv2 3. OSPFv2 Route Summarization and Filtering 4. Implement Multiarea OSPFv3
7	Implement BGP Communities 1. Implement MP-BGP 2. Implement eBGP for IPv4
8	Implement IPsec Site-to-Site VPNs 1. Implement GRE over IPsec Site-to-Site VPNs 2. Implement VRF Lite
9	Simulating SDN with 1. OpenDaylight SDN Controller with the Mininet Network Emulator 2. OFNet SDN network emulator
10	Simulating OpenFlow Using MININET

Minor- Research Information Systems and Computing

COURSE CODE: **COURSE CREDIT: 04**

1 credit - 15 lectures

1 lecture - 60 minutes

Course Objectives:

- The research methodology course is proposed to assist students in planning and carrying out research projects. The students are exposed to the principles, procedures and techniques of implementing research project.
- The course starts with an introduction to research and carries through the various methodologies involved. It continues with finding out the literature using technology, basic statistics required for research and finally report writing.
- Define research, formulate problem and describe the research process and research methods.
- Understand and apply basic research methods including research design, data analysis and interpretation, understand ethical issues in research, write research report, research paper and publish the paper.

UNIT NO	SYLLABUS	NO. OF LECTURES
01	**Introduction:** Role of Business Research, Information Systems and Knowledge Management, Theory Building, Organization ethics and Issues. **Beginning Stages of Research Process:** Scientific Research: Definition, Characteristics, types, need of research. Identification of the problem, assessing the status of the problem, formulating the objectives, preparing design (experimental or otherwise), Actual investigation, Determining the mode of attack. Problem definition, Qualitative research tools, Secondary data research. **Literature survey:** References, Abstraction of a research paper, Possible ways of getting oneself abreast of current literature.	15
02	**Research Methods and Data Collection:** Survey research, communicating with respondents, Observation methods, Experimental research. **Measurement Concepts, Sampling and Field work:** Levels of Scale measurement, attitude measurement, questionnaire design, sampling designs and procedures, determination of sample size. **Data Analysis and Presentation:** Editing and Coding, Basic Data Analysis, Univariate Statistical Analysis and Bivariate Statistical analysis and differences between two variables. Multivariate Statistical Analysis.	15
	Documentation and scientific writing: Results and Conclusions, Preparation of manuscript for Publication of Research paper, Presenting a paper in	

03	scientific seminar, Thesis writing. Structure and Components of Research Report, Types of Report: research papers, thesis, Research Project Reports, Pictures and Graphs, citation styles, writing a review of paper, Bibliography. **Publishing the Paper:** Rights and Permissions, How to Submit the Manuscript, How and When to Use Abbreviations, How to Write a thesis, Outcome of Research, Ethical issues in research	**15**
04	**Statistical analysis and fitting of data :**Introduction to Statistics – Probability Theories - Conditional Probability, Poisson Distribution, Binomial Distribution and Properties of Normal Distributions, Estimates of Means and Proportions; Chi-Square Test, Association of Attributes - t-Test –Anova- Standard deviation - Co-efficient of variations. Co- relation and Regression Analysis.	**15**

Textbook(s):

1. Business Research Methods William G.Zikmund, B.J Babin, J.C. Carr, Atanu Adhikari, M.Griffin Cengage 8e 2016.
2. Business Analytics Albright Winston Cengage 5e 2015.
3. Research Methods for Business Students Fifth Edition Mark Saunders 2011.
4. Multivariate Data Analysis Hair Pearson 7e 2014.
5. Thesis & Assignment Writing–J Anderson, B.H. Dursten &M.Poole, Wiley Eastern, 1977
6. A Hand Book of Methodology of Research – P. Rajammal and P. Devadoss, R. M. M. Vidya Press, 1976.

SIES NERUL COLLEGE OF ARTS SCIENCE AND COMMERCE(Autonomous)
NEP STRUCTURE 2026-27
M Sc Computer Science

Sem	Major		RM	OE	VSC, SEC	IKS, AEC, VEC	OJT, FP	RP	Cum. Cr./ Sem.
	Mandatory	Elective (Any One) Through NPTEL							
II	Applied Machine Learning (3+1P)Credit	Bioinformatics 04 Credit OR Embedded and IoT Technology 04 Credit		-	-	-	04		22
	Compiler Designing (3+1P) Credit								
	Software Architecture and Design pattern (3+1P) Credit								
	Cyber Law and Ethics (2 Credit)								
	14	04					04		22

Major - Applied Machine Learning

COURSE CODE : **COURSE CREDIT: 03**

1 credit - 15 lectures

1 lecture - 60 minutes

Course Objectives:

- To understand the importance and concepts of Machine Learning.
- Knowledge on various learning techniques, performance measure, and Artificial Neural Network (ANN).
- Applying algorithms available for the machine learning.
- Designing and developing practical Machine Learning and Machine Learning based applications.

Course Outcomes:

- The ability to describe the basic pragmatics of machine learning.
- Discover various projects in machine learning for industrial applications.
- Assess and Evaluate semantics of ML based systems.
- Analyse Machine Learning with visualization techniques and develop real-time applications.

Unit	Syllabus	No. of lectures
1.	**The Fundamentals of Machine Learning** What is Machine Learning? Why use Machine Learning? Types of Machine Learning, Supervised Learning, Unsupervised Learning & Reinforcement Learning. Challenges of Machine Learning, Testing and Validation, A First Application: Classification, MNIST Dataset, Performance Measures, Confusion Matrix, Precision and Recall, Precision/Recall Tradeoff, The ROC Curve, Multiclass Classification, Error Analysis. **Case Study**	15
2.	**Regression Analysis** Linear Regression, Gradient Descent, Batch Gradient Descent, Stochastic Gradient Descent, Mini-batch Gradient Descent, Polynomial Regression, Learning Curves, The Bias/Variance Tradeoff, Ridge Regression, Lasso Regression, Early Stopping, Logistic Regression, Decision Boundaries, Softmax Regression, Cross Entropy.	15

	Case Study	
3.	**Supervised Learning** Linear SVM Classification, Soft Margin Classification, Nonlinear SVM Classification, Polynomial Kernel, Gaussian RBF Kernel, SVM Regression, Decision Trees, Training and Visualizing a Decision Tree, Making Predictions, The CART Training Algorithm, Gini Impurity vs Entropy, Regularization Hyperparameters. **Case Study**	15

References:

1. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems by AurélienGéron, Second Edition, O'reilly 2019

2. Deep Learning with Python by François Chollet Published by Manning 2018

3. Reinforcement Learning: An Introduction by Richard S. Sutton and

Andrew G. Barto, Second Edition 2014

Additional References:

1. Introduction to Machine with Python - A Guide for Data Scientists by Andreas C. Müller & Sarah Guido O'reilly 2016

Major- Applied Machine Learning Practical

COURSE CODE : **COURSE CREDIT: 01**

1 credit - 30 lectures

1 lecture - 60 minutes

Sr. No.	List of Practical
1	Write a program to implement Linear Regression (Diabetes Dataset)
2	Write a program to Implement Logistic Regression (Iris Dataset)
3	Write a program to Implement Multinomial Logistic Regression (Iris Dataset)
4	Write a program to Implement SVM classifier (Iris Dataset)
5	Write a program to Train and fine-tune a Decision Tree for the Moons Dataset
6	Write a program to Train an SVM regressor on the California Housing Dataset
7	Write a program to Implement Batch Gradient Descent with early stopping for Softmax Regression
8	Write a program to Implement MLP for classification of handwritten digits (MNIST Dataset)
9	Write a program to implement SVM classification on the Iris dataset.
10	Mini Project

Compiler Designing

COURSE CODE : **COURSE CREDIT: 03**

1 credit - 15 lectures

1 lecture is 60 minutes

Course Objectives:

- Provide an understanding of the fundamental principles in compiler design.
- Provide the skills needed for building compilers for various situations.
- Learn the process of translating a modern high-level language to executable code required for compiler construction.

Course Outcomes:

- Familiar with fundamentals of compiler and identify the relationships among different phases of the compiler.
- Interpret the application of finite state machines, recursive descent, production rules, parsing, and language semantics.
- Analyze& implement required module, which may include front-end, back-end, and a small set of middle-end optimizations.

Unit	Syllabus	No. of lectures
1.	**Introduction, Lexical Analysis** Language Processors, The structure of a Compiler, The evolution of Programming Languages, The science of building a Compiler, Applications of Compiler Technology, Programming Language Basics Role of Lexical Analyzer, Specification of Tokens, Token Recognition, Nondeterministic Finite Automata (NFA), Deterministic Finite Automata (DFA), NFA to DFA, Regular Expression to NFA, Optimization of DFA, DFA Minimization Algorithm **Case Study:** Lexical Analysis Tool-Lex	15
2.	**Syntax Analysis, Syntax Directed Translation** Role of Parser, Error Handling, Grammar, Top-Down Parsing, Bottom-up Parsing, LR Parsing, What is SDT, Attributes, Evaluating Attributes in SDD, Classes of SDD, Side Effects in SDD, Syntax-Directed Translation (SDT) Schemes, Postfix SDT, Infix SDT, Converting SDD to SDT **Case Study:** LALR Parser Generator-yacc	15
3.	**Type Checking, Symbol Tables and Runtime Environment Management** Static vs. Dynamic Checking, Type Expressions, Type Checking, Type Equivalence, Type Conversion Information in Symbol Table, Operations on Symbol Tables, Simple & Scoped Symbol Table Introduction, Activation Record, Environment without Local Procedures, Environment with	15

	Local Procedures, Display **Case Study**	

References:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, J. D. Ullman, "Compilers: Principles, Techniques, & Tools", 2nd Edition, Pearson Education, 2013, ISBN: 978-9332518667.

2. Santanu Chattopadhyay, "Compiler Design", PHI Learning Pvt. Ltd., 2015.

Additional References:

1. Dick Grune, Henry E. Bal, Ceriel J. H. Jacobs, Koen G. Langendoen, "Modern Compiler Design", John Wiley & Sons Publisher, 2000, ISBN: 978-0471976974

2. Barret, Couch, Compiler Construction Theory and Practice, , Asian Student Edition.

3. Dhamdhere D.M, "Compiler Construction Principle and Practice", McMillan India;

4. David Galles, Modern Compiler Design, Pearson Education, 2009

Compiler Designing - Practical

COURSE CODE : **COURSE CREDIT: 01**

1 credit - 30 lectures

1 lecture is 60 minutes

Sr. No.	List of Practical
1	Program to count the number of vowels and consonants in a given string.
2	Program to count the number of characters, words, spaces, end of lines in a given input file.
3	Program to count no of: a) +ve and –ve integers b) +ve and –ve fractions
4	Program to count the no of comment line in a given C program. Also eliminate them and copy that program into separate file
5	Program to count the no of 'scanf' and 'printf' statements in a C program. Replace them with 'readf' and 'writef' statements respectively.
6	Program to recognize a valid arithmetic expression and identify the identifiers and operators present. Print them separately.
7	Program to recognize whether a given sentence is simple or compound.
8	Program to recognize and count the number of identifiers in a given input file.
9	Program to test the validity of a simple expression involving operators +, -, * and /.
10	Program to recognize nested IF control statements and display the levels of nesting.
11	Program to check the syntax of a simple expression involving operators +, -, * and /.
12	Program to recognize a valid variable, which starts with a letter, followed by any number of letters or digits.
13	Program to evaluate an arithmetic expression involving operating +, -, * and /.
14	Program to recognize strings 'aaab', 'abbb', 'ab' and 'a' using grammar (a^nb^n, n>=0)
15	Program to recognize the grammar (a^nb, n>=10)

MAJOR- Software Architecture and Design Pattern

COURSE CODE: **COURSE CREDIT: 03**

1 credit – 15 lectures

1 lecture - 60 minutes

Course Objectives:

- To provide students with a structured understanding of software engineering principles, process models, and architectural concepts used in modern software development.
- To develop competence in software testing techniques, quality assurance practices, and testing strategies applicable to real-world software systems.
- To introduce architectural frameworks and MVC-based application development using industry-relevant frameworks such as Django and Struts.

Course Outcomes:

After successful completion of the course, students will be able to:

- Apply software engineering concepts, process models, and design principles to analyze and design software systems.
- Design and execute appropriate software testing strategies, prepare test artifacts, and evaluate software quality.
- Develop and test web applications using MVC architecture and framework-based approaches such as Django and Struts.

Unit No.	Syllabus	No. of Lectures
I	Software Engineering: Software engineering concepts, software development life cycle, software process models: waterfall model, incremental model, spiral model, agile process models, requirement engineering, feasibility study, software project planning, software cost estimation techniques: LOC, Function Point Analysis, software design principles, modular design, cohesion and coupling, software architecture concepts, software configuration management, version control, change management.	15
II	**Software Testing and Quality Assurance:**Software testing fundamentals, levels of testing: unit testing, integration testing, system testing, acceptance testing, testing techniques: black-box testing, white-box testing, gray-box testing, test case design techniques, test plan preparation, defect life cycle, debugging techniques, software quality concepts, software reliability, verification and validation, regression testing, introduction to automation testing, testing of web-based applications.	15
III	Software Architecture and Application Frameworks: Software architecture for web applications, Model–View–Controller (MVC) architecture, MVC components and flow, web application request–response lifecycle, Django framework: architecture, models, views, templates, URL routing, ORM, Struts framework: architecture, components, MVC implementation in Struts, comparative study of MVC	15

	frameworks, deployment and security basics.	

Textbooks:

1. Software Engineering: A Practitioner's Approach – Roger S. Pressman
2. Software Testing – Naresh Chauhan
3. Django for Beginners – William S. Vincent

Reference(s):

1. Software Engineering – Ian Sommerville
2. Foundations of Software Testing – Rex Black, Erik van Veenendaal, Dorothy Graham
3. Agile Testing – Lisa Crispin and Janet Gregory
4. Beginning Struts 2 – Arnold Doray

Software Architecture and Design Pattern Practical

COURSE CODE: **COURSE CREDIT: 01**

1 credit - 30 lectures

1 lecture - 60 minutes

Practical No.	List of Practical
1	Study of software development life cycle models used in real-world projects.
2	Preparation of software project estimation using Function Point or LOC methods.
3	Conduct a test suite for two different websites using Selenium IDE. Perform various actions like clicking links, filling forms, and verifying content.
4	Write a program using Selenium WebDriver to automate the login process on a specific web page. Verify successful login with appropriate assertions.
5	Write a program using Selenium WebDriver to update 10 student records in an Excel file. Perform data manipulation and verification.
6	Preparation of test plan and test cases for a given software application.
7	Study and implementation of MVC architecture using a simple application.
8	Development of a basic web application using Django framework.
9	Implementation of MVC-based application using Struts framework.
10	Testing and validation of the developed web application.

Major: Cyber Law and Ethics

COURSE CODE : **COURSE CREDIT: 02**

1 credit - 15 lectures

1 lecture - 60 minutes

Course Objectives:

- Provide fundamental understanding of cyber laws, cyber technologies, and network security.
- Introduce the legal framework governing cyberspace in India, including the Information Technology Act, 2000.
- Explain concepts of cryptography, encryption techniques, and digital and electronic signatures.
- Familiarize students with key issues in cyber laws such as e-commerce, e-governance, e-records, and e-contracts.

Course Outcomes:

- Explain basic concepts of cyber laws, internet technology, networks, and network security.
- Describe the legal framework of cyber law in India, including the provisions of the Information Technology Act, 2000.
- Understand and apply concepts of cryptography, encryption techniques, digital signatures, and electronic signatures.
- Analyze legal issues related to e-commerce, e-governance, e-records, e-contracts, and the role of regulators and certifying authorities..

Unit No.	Syllabus	No. of lecture
I	Introduction to Cyber Laws and Technology: Basic Concepts, Internet and Advantages and Disadvantages of Internet Technology, Network and Network Security. Legal Framework and Regulations: Cyber Law & Components of Cyber Law, Cyber Law in India: An Overview of Information Technology Act2000, Cryptography, Encryption Technique & Algorithm and Digital Signature & Electronic Signature Key Issues in Cyber Laws: E-Commerce, E-Governance, E-Record & E- Contract, Regulator, Certifying Authority, Electronic Signature Certificates	15

II	Introduction to Cyber Ethics: Concept and importance of cyber ethics; relationship between cyber law, ethics, and technology. Ethics in Internet and Network Technology: Ethical use of internet; advantages and disadvantages of internet technology; responsible use of networks; network security ethics; ethical and unethical hacking. Privacy and Data Protection Ethics: Privacy, confidentiality, data protection, surveillance, and information integrity in cyberspace. Ethical Issues in Cyber Applications: Ethics in e-commerce, e-governance, e-records and e-contracts; trust, transparency, and consumer protection. Ethics in Digital Security and Regulation: Ethical aspects of cryptography and encryption; digital and electronic signatures; role and ethical responsibility of regulators and certifying authorities; electronic signature certificates.	15

References:

1.Yatindra Singh, *Cyber Laws*, Universal Law Publishing, New Delhi.

2.Pavan Duggal, *Cyber Law: The Indian Perspective*, Saakshar Law Publications.

3.Rodney D. Ryder, *Guide to Cyber Laws*, Wadhwa & Company.

4.Tavani, Herman T., *Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing*, Wiley.

5.Information Technology Act, 2000 and Information Technology (Amendment) Act, 2008.

ELECTIVE - Bioinformatics

COURSE CODE: **COURSE CREDIT: 03**

1 credit - 15 lectures

1 lecture is 60 minutes

Pre requisite: Basic concepts of Biology, Data Structures and Algorithms, Data Analysis and

Visualization.

Course Objectives:

- Gain a solid understanding of fundamental concepts and principles in bioinformatics,including sequence analysis, genome analysis, protein structure prediction, and gene expression analysis
- Acquire the ability to analyze and interpret biological data, such as DNA or protein sequences, microarray data, or next-generation sequencing data
- Learn statistical methods and techniques for extracting meaningful insights from large datasets.

Course Outcomes:

After successful completion of this course students will

- Develop skills in using bioinformatics tools and software packages commonly used in the field, such as BLAST, EMBOSS and Biopython
- Awareness of Ethical and Legal Considerations Develop the ability to stay updated with the latest advancements and emerging trends in bioinformatics research and technologies

Unit	Syllabus	No. of lectures
01	Biological Data Analysis Biological Foundations: Introduction to molecular biology concepts and terminology, DNA, RNA, and protein structure and function, Genetic variation and mutation Introduction to Bioinformatics: Overview of bioinformatics and its applications in biology and medicine, Introduction to biological databases and data formats, Introduction to sequence analysis, structure analysis Sequence Analysis: Sequence alignment algorithms (pairwise and multiple sequence alignment), Sequence database searching (BLAST, FASTA), Hidden Markov Models (HMMs) for sequence analysis, Phylogenetic analysis and evolutionary tree construction Case Study	15

02	Structure Analysis: Protein structure prediction methods (homology modeling, ab initio methods), Protein structure visualization and analysis tools, Drug discovery Computational Tools and Methods Genomics and Transcriptomics: Analyzing and manipulating genomic sequences, working with genome annotations and gene features, Analyzing gene expression data (RNA-Seq, microarray), Identifying differentially expressed genes Data Visualization and Reporting: Visualizing bioinformatics data, Creating interactive visualizations of biological data Case Study	15
03	Analytics and Business: Introduction to machine learning algorithms and techniques, Feature selection and dimensionality reduction in biological data, Predictive modeling for biological data (classification, regression) Ethical, Legal, and Social Implications: Ethical considerations in bioinformatics research, Privacy and data security in genomic data, social and policy issues in bioinformatics and personalized medicine Case Study	15

References:

1. Bioinformatics: Sequence and Genome Analysis by David W. Mount Publisher: Cold Spring Harbor Laboratory Press Publication (4th edition), 2021,
2. Python for Bioinformatics by Tiago Antao, Packt Publishing Publication, 2015
3. Python for Biologists: A complete programming course for beginners" by Martin Jones CreateSpace Independent Publishing Platform, 2013,
4. Bioinformatics for Beginners: Genes, Genomes, Molecular Evolution, Databases, and Analytical Tools by Supratim Choudhuri, Academic Press Publication, 2014
5. Bioinformatics Programming Using Python: Practical Programming for Biological Data by Mitchell L. Model, O'Reilly Media, 2009

Bioinformatics Practical

COURSE CODE: **COURSE CREDIT: 01**

1 credit - 2 lectures **1 lecture is 60 minutes**

Sr.No.	List of Practicals
1	Sequence Manipulation • Read and parse sequence data from files • Perform basic sequence manipulations (e.g., reverse complement, translation)
2	Sequence Alignment • Perform pairwise sequence alignment using algorithms like Needleman- Wunsch or Smith-Waterman • Implement multiple sequence alignment using methods such as ClustalW or MUSCLE
3	Database Searching • Perform sequence searches against databases (e.g., BLAST or FASTA) • Retrieve and analyze search results
4	Protein Structure Analysis • Retrieve protein structures from databases like PDB • Calculate structural properties (e.g., secondary structure, solvent accessibility) • Perform structure visualization and analysis
5	Genomic Data Analysis • Retrieve genomic data from databases (e.g., NCBI) • Analyze gene annotations, promoter regions, or regulatory elements • Perform genomic variant analysis
6	Data Preprocessing • Cleaning and preprocessing biological data (e.g., gene expression data, DNA sequences) • Handling missing values, outliers, and normalization of data • Feature selection and dimensionality reduction techniques
7	Classification • Applying machine learning algorithms (e.g., decision trees, random forests, support vector machines) to classify biological samples or sequences • Evaluating model performance using metrics such as accuracy, precision, recall, and F1-score
8	Regression • Building regression models to predict quantitative biological properties (e.g., protein structure, gene expression levels) • Assessing model performance using metrics such as mean squared error or R-squared
9	Clustering • Applying clustering algorithms (e.g., k-means, hierarchical clustering) to group similar biological samples or sequences • Assessing clustering quality using metrics such as silhouette coefficient or Rand index
10	Visualizing clusters and analyzing their biological significance • Data Visualization: • Generate plots, graphs, and figures to visualize bioinformatics results • Use libraries like Matplotlib, Seaborn, or ggplot in Python or R for visualization • Create interactive visualizations using tools like D3.js or Plotly

Major Elective-Embedded and IoT Technology

COURSE CODE: **COURSE CREDIT: 03**

1 credit - 15 lectures

1 lecture is 60 minutes

Pre requisite: Basic electronics knowledge (components, microcontrollers), understanding of

wireless sensor networks, familiarity with IoT concepts and architectures.

Course Objectives:

- Understand embedded systems design and basics of IoT components.
- Learn about electronics and microcontrollers in embedded systems.
- Gain knowledge of IoT building blocks, sensors, and wireless sensor networks.
- Explore advanced IoT technologies, including gateway architecture and cloud computing.

- Understand IoT security, communication, and design connected IoT systems using

Packet Tracer.

Sr. No	Syllabus	No. of lectures
01	Module I Embedded System Basics Introduction to Embedded Systems- Design of Embedded Systems, Memory Architecture, Input/Output. Basic electronics: Semiconductors, Transistors, BJT, Flip Flops, Resistors, Capacitors. Microcontrollers, UART Communications, SPI-peripherals interface, I2C communication, Wireless Sensor Network (WSN) Case Study	15
02	Module II Basics of IOT- Introduction IoT, IoT Building Blocks -Hardware and Software: The basic IoT building blocks, smart thing components and capabilities, basics of Packet Tracer with reference to IoT, basics of IoT gateway, Cloud, and analytics IoT Gateway:	15

	IoT architecture domains, IoT gateway architecture, IoT gateway functionalities, IoT gateway selection criteria, IoT gateway and edge computing, edge computing-based solution for specific IoT applications IoT Protocol Stack, IoT Cloud and Fog Computing: Components of IoT Cloud architecture, usage of application domains of IoT Cloud platforms, layered architecture of Fog computing, distinguish Fog computing from other related terms. Case Study	
03	Module III Security, Communication and Data analytics in IOT-IoT Security: Security constraints in IoT systems, security requirements of IoT systems, IoT attacks, security threats at each layer of IoT architecture, design secure IoT system for specific application Social IoT: Nature of social relationships among IoT Devices, functionality of different components of social IoT architecture, social aspects of smart devices in IoT applications Packet Tracer and IoT: Basics of Packet Tracer and Blockly programming language, design simple IoT projects in Packet Tracer. Case Study	15

References:

1. Introduction to Embedded Systems – Cyber physical systems Approach Edward AshfordLee & Sanjit Arun kumarSeshia Second Edition — MIT Press — 2017
2. Enabling the Internet of Things Fundamentals, Design and Applications by MuhammadAzhar Iqbal, Sajjad Hussain, Huanlai Xing, Muhammad Ali Imran Wiley Pub.1st Edition2021
3. Introduction Embedded Systems by K.V. Shibu Second Edition McGraw Hills–2017
4. Build your own IoT Platform Develop a Fully Flexible and Scalable Internet of ThingsPlatform in 24 Hours by AnandTamboli, 2019 ,Apress

Embedded and IoT Technology Practical

COURSE CODE:
01

COURSE CREDIT:

1 credit - 2 lectures

1 lecture is 60 minutes

Note: - The following set of practicals should be implemented in CodeVisionAVR, Proteus8, Cisco Packet Tracer, Keli V5, Python Link: - Python:https://www.python.org/downloads/ CodeVisionAVR https://www.codevision.be/Proteus8:https://www.labcenter.com/downloads/ Cisco Packet Tracer:https://www.netacad.com/courses/packet-tracer,Keli V5: https://www.keil.com/download

Sr. No.	List of Practicals
1	Design and implement basics embedded circuits 1. Automatic Alarm system- Alarm should get trigger by senor 2. Timer based buzzer 3. Sensor based Counting device
2	Demonstrate communication between two embedded devices using UART port
3	Built an IoT system to send ticket before entering the bus.
4	Demonstrate an IoT based game which can be played between two player who are physically at a considerable distance.
5	Develop a IoT application which will record the movement and orientation of your phone and give the data back to the PC
6	Develop an IoT application that will raise an alarm whenever with going to rain outside based on the weather prediction data.
7	Deploy an IoT application which will alert you by beeping or vibrating your phone whenever you get someone call your name.
8	Develop an IoT application for monitoring water levels in tanks and automatically start the motor to fill the tank if the level goes below the critical level.
9	Develop an IoT module to which measure the intensity of light and send the same to your PC/ Phone
10	Develop an IoT application for Motion detection.

SCHEME OF THEORY and PRACTICALS EXAMINATION

4 CREDIT PAPERS

The scheme of examination shall be divided into two parts:

- Internal assessment 40% i.e. 40 marks
- Semester end examination 60% i.e. 60 marks

A. Internal Assessment 40 marks

Description	Marks
Internal test of 20 marks 1. Multiple choice Questions / True or False - 20 Marks	20
One Project and Viva voce / Presentation / Case studies / Poster Making / Quiz / Role Play / Subject Specific Activities	15
Attendance and Class behavior	05
Total	40

B. Semester end examination 60 marks

PAPER PATTERN

Duration: 2 hours			
Total Marks: 60			
All Questions are Compulsory			
Question	**Based on**	**Options**	**Marks**
Q. 1	Unit 1	A and B or P and Q	15
Q. 2	Unit 2	A and B or P and Q	15

Q. 3	Unit 3	A and B or P and Q	15
Q. 4	Unit 1,2,3	A and B or P and Q	15
Total			60

Note:

Q.1, 2, 3 and 4 may be divided into sub questions with internal choice if required.

- Passing criteria: Minimum 40% in Internal (16 out of 40) and 40% (24 out of 60) in semester end examination.

SCHEME OF PRACTICAL EXAMINATION

The scheme of Practical examination shall be

- Practical assessment carries 50 Marks : 40 marks + 05marks (journal)+ 05 marks(viva)
- Minimum 75 % practical are required to be completed and written in the journal. (Certified Journal is compulsory for appearing at the time of Practical Exam)

(A) Practical Assessment 50 marks

Description	**Marks**
Two questions of practical (20 marks each)	40
Journal	5
Viva	5
Total	50

- **Passing criteria: Minimum 40% in Practical (20 out of 50)**

SCHEME OF EXAMINATION FOR MAJOR PAPERS WITH 2 CREDITS

The scheme of examination shall be :

(A) **Internal Assessment 20 marks**

Description	Marks
Internal test of 10 marks (Multiple choice Questions / True or False)	10
One Project and Viva voce / Presentation / Case studies / Poster Making / Quiz / Role Play / Subject Specific Activities	05
Attendance and Class behavior	05
Total	20

Passing criteria: Minimum 40% in Internal (8 out of 20)

B. Semester end examination 30 marks

PAPER PATTERN

Duration: 1 hours			
Total Marks: 30			
All Questions are Compulsory			
Question	**Based on**	**Options**	**Marks**
Q. 1	Unit 1	A and B or P and Q	10
Q. 2	Unit 2	A and B or P and Q	10
Q. 3	Unit 1,2	A and B or P and Q	10
Total			30

Note:

Q.1, 2 and 3 may be divided into sub questions with internal choice if required.

- Passing criteria: 40% (12 out of 30) in semester end examination.