Four Year Under Graduate programme (FYUP) Under NEP 2020 w.e.f. 2025-26 academic session

Course Structure and Credit Allocation (For Subjects with Minimal or No Practical Component)

The following course structure under FYUP is designed for subjects which do not have practical based courses or have minimal offerings of practical course-based learning:

First Year (NHEQF Level- 4.5)

Course Category	Semester-I					Semester-II			
	Subject/Title	No. of paper	Credits			Subject /Title	No. of paper	Credits	
			T	P				T	P
Major	Core Subject-I Programming in C	1	4	2		Core Subject-I Data Structure Using C	1	4	2
	Core Subject-II	1	4	2		Core Subject-II	1	4	2
Minor	M.D-I/ I.D-I FIT	1	2	2		M.D-II/ I.D-II IOT	1	2	2
SEC	SEC-I (Any one) 1. Office Automation 2. Web development basic	1	2	-		SEC-II (Any one) 1. Office Automation 2. Web development basic	1	2	-
VAC	Understanding and connecting with environment Or Life Skills & personality development	1	2	--		Understanding and connecting with environment Or Life Skills & personality development	1	2	-
Total		5	14	6			5	14	6
NHEQF Level 4.5	*Student on exit after successfully completing first year (i.e., securing minimum required 40 credits + 4 Credits in* one vocational course/skills-enhancement course of 4 credits*) will be awarded "Undergraduate Certificate" of one year, in related field/discipline/subject*								

Course: Programming in C (SOS/CS/MJ/C101)

Course Objective:

The Programming in C course aims to provide students with a solid foundation in programming using the C language. Participants will start by learning the basics of C syntax, data types, and control structures, progressing to more advanced topics like functions, pointers, and memory management. The course focuses on developing problem-solving skills through hands-on coding exercises and projects, enhancing participants' ability to translate real-world problems into C programs.

Course Outcome:

By the end of this course, students should be able to:

1. Understand the basic programming concepts and syntax of the C language.

2. Design and implement algorithms to solve simple programming problems.

3. Write, compile, and execute C programs using integrated development environments (IDEs).

Course Content/Syllabus:

Unit-1: Introduction to C Programming Basics of C programming language, Structure of C program, Data types, Variables, Constants, Input and Output functions, Operators, Expressions, Control structures: if, if-else, switch.

Unit-2: Loops and Functions Looping constructs: while, do-while, for, Nested loops, Arrays, Strings, Pointers, Functions: declaration, definition, call, return, recursion.

Unit-3: Storage Classes and Preprocessor Storage Classes: auto, extern, static, register, Scope and lifetime of variables, Introduction to the Preprocessor, Macros, Conditional compilation.

Unit-4: User-defined Data Types Structures and Unions, Introduction to user-defined data types, Structures: declaration, initialization, accessing members, Arrays of structures, Pointers to structures, Unions: declaration, accessing members.

Unit-5: File Handling and Command-line Arguments File operations: Opening, Reading, Writing, and Closing files, Random Access to files, Command- line arguments, File handling functions, Error handling.

Recommended Books:

1. "C Programming Absolute Beginner's Guide" by Greg Perry and Dean Miller.

2. "The C Programming Language" by Brian W. Kernighan

Course: Fundamentals of Information technology (SOS/CS/MI/1.1) (SOS/CS/MD102)

Course Objective:

1. Take on significant positions in various IT work

2. Collaborate in diverse team environments

3. Contributions in the field of IT

4. Work effectively in the IT field to make a positive contribution to society

Course Outcome:

1. Develop information technology solutions by evaluating user requirements in the systems development environment.

2. Apply knowledge of IT requirements for technology solutions in cutting edges applications.

3. Analyze a problem and identify and define the computing requirements for the appropriate solutions.

4. Create, select and apply appropriate techniques, resources, and modern engineering and IT tools.

Course Content:

Unit 1. Introduction: Definition of Electronic Computer, Generations, Classification of Computers, Computer Hardware and Basic Computer Organization: CPU- ALU, CU; RAM/ROM, Various I/O devices, Peripherals, Storage Media.

Unit 2. Computer Languages: Binary, Hexadecimal Number System; Basic Binary Logic Operations; Binary Addition and Subtraction; Generation of Languages, Assembly Language, High level language; Translators, Interpreters, Compilers, Compilers; Flow Charts, Dataflow Diagram,

Unit 3. OS & Office: Software- System and Application Software; Elementary Concepts in Operating System; Textual Vs GUI, Introduction to DOS, MS Windows, UNIX/Linux

Unit 4. Computer Networks: Elements of Communication system; Brief Introduction to Computer Networks- Introduction of LAN and WAN. Network Topologies, Client-server Architecture, IoT, Cloud Computing

Unit 5. Internet: Internet & World Wide Web, Hypertext Markup Language, DHTML, Python, WWW, Gopher, FTP, Telnet, Web Browsers, Net Surfing, Search Engines, Email; Introduction to Web Development, Static and Dynamic Pages

Unit 6. IT Application and Multi-media: Basic Awareness of NICNET and ERNET; E Commerce, E governance; Brief Introduction to Different Formats of Image, Audio, Video

Unit 7. Information Concepts & Processing Definitions of Information, Need of information, quality of information, value of information, concept of information, Entropy category and Level of information in Business Organization, Data Concepts and Data Processing, Data Science, Data Representation, Application of IT to E-commerce, Electronic Governance, Multimedia, Entertainment, Introduction to Information System.

Recommended Books:

1. Sinha, Sinha, "Computer Fundamentals",

2. Yadav R. P. , "Information Technology"

Course: Office Automation (SOS/CS/SEC103)
Course Objective: The "Office Automation" course aims to familiarize participants with tools and techniques for streamlining office tasks using digital technologies. By the course's end, students will be proficient in using office software applications for document creation, data analysis, and communication. The objective is to enhance productivity, collaboration, and efficiency in modern office environments through effective utilization of automation tools.
Course Outcome: By the end of this course, students should be able to: **1.** Effectively use office automation tools for document creation, presentation, and data analysis. **2.** Apply advanced features of office software to optimize office tasks and workflows. **3.** Demonstrate effective communication and collaboration skills using office automation tools.
Course Content/Syllabus: **Unit-1:** Introduction to Office Automation Overview of office automation and its benefits, Office Suite applications and their features. **Unit-2:** Word Processing and Spreadsheets Word processing: Formatting, Styles, Mail Merge, Spreadsheets: Formulas, Functions, Charts. **Unit-3:** Presentation Tools and Email Management Creating and delivering presentations, Email and Calendaring: Managing emails and appointments. **Unit-4:** Database Management and Workflow Automation Database management: Creating and querying databases, Document collaboration and version control, Workflow automation using macros and scripts.
Recommended Books: **1.** "Microsoft Office 365 & Office 2019: Introductory" by Misty E. Vermaat. **2.** "Office 2019 All-in-One for Dummies" by Peter Weverka.

Course: Web Development Basics (SOS/CS/SEC104)

Course Objectives

1. Understand the fundamentals of HTML for creating structured web pages.
2. Learn CSS for styling and layout of web pages.
3. Build static, responsive, and visually appealing websites.
4. Gain a foundation for further learning in JavaScript and advanced web technologies.

Course Outcomes

1. Create structured web pages using HTML tags.
2. Apply CSS properties for styling, colors, and layouts.
3. Design responsive web pages using CSS positioning and media queries.
4. Develop a static website combining HTML and CSS effectively.

Unit	Topics
Unit 1: Introduction to Web Development	Basics of the Internet, WWW, Browser, Web Servers, Overview of HTML, CSS, and their roles in web development.
Unit 2: HTML Basics	Structure of HTML document, Basic Tags (`<html>`, `<head>`, `<title>`, `<body>`), Text formatting (`<h1>`–`<h6>`, `<p>`, `<b>`, `<i>`, `<u>`), Lists, Links, Images.
Unit 3: Tables and Forms in HTML	Creating tables (`<table>`, `<tr>`, `<td>`, `<th>`), Table attributes, Merging cells, Creating forms (`<form>`, `<input>`, `<textarea>`, `<select>`, `<button>`).
Unit 4: CSS Fundamentals	Introduction to CSS, Inline, Internal, and External CSS, Syntax and Selectors (element, class, id), Colors, Fonts, Text Styling, Box Model (margin, padding, border).
Unit 5: CSS Layouts and Positioning	Background properties, Display property, Positioning (static, relative, absolute, fixed), Flexbox, Grid Layout, Navigation bars, Z-index.
Unit 6: Responsive Web Design	Media Queries, Viewport meta tag, Relative units (`em`, `%`, `vh`, `vw`), CSS units for responsiveness.
Unit 7: Advanced CSS Features	CSS Transitions, Animations, Pseudo-classes (`:hover`, `:nth-child`), Pseudo-elements (`::before`, `::after`), CSS Variables.

Recommended Books:

1. HTML & CSS: Design and Build Websites – Jon Duckett
2. Learning Web Design – Jennifer Niederst Robbins

Course: Data Structure (SOS/CS/MJ/C201)
Course Objective: The objective of this course is to familiarize students with fundamental data structures and their applications in solving computational problems. The course aims to develop students' skills in designing, implementing, and analysing various data structures
Course Outcome: By the end of this course, students should be able to: **1.** Understand the concepts of data structures and their importance in computing. **2.** Implement and use common data structures such as arrays, linked lists, stacks, queues, trees, and graphs. **3.** Analyze the time and space complexity of algorithms related to data structures. **4.** Apply data structures to efficiently solve real-world problems.
Course Content: **Unit-1:** Introduction to Data Structures Overview of data organization and data types, Arrays, Linked Lists, and Stacks, Queues and their implementations. **Unit-2:** Trees and Graphs Binary Trees and Binary Search Trees, AVL Trees and Red-Black Trees, Graph representation and traversal. **Unit-3:** Hashing Introduction to hashing and its applications. Hash functions and collision resolution techniques. Hash tables and their implementation. Understanding the efficiency of hash-based data structures. **Unit-4:** Sorting and Searching Algorithms Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, Heap Sort, Linear and Binary Search. **Unit 5:** File Organization and Indexing Fundamentals of file organization: sequential, direct, and indexed. Primary and secondary indexing techniques. B+-tree index structure and its role in efficient data retrieval. Case studies of file organization in database systems.
Recommended Books: **1.** "Data Structures and Algorithms in Java" by Robert Lafore. **2.** "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.

Course: Internet of Things (SOS/CSE/MI/E2.1) (SOS/CSE/MD202)

Course Objective:

The Internet of Things (IoT) course aims to provide students with a comprehensive understanding of the IoT ecosystem, including its architecture, design principles, and practical implementations. The course introduces core concepts such as sensing, actuation, networking, and data management, while emphasizing the role of cloud computing, security, and real-world applications. Students will gain hands-on experience with hardware platforms like Arduino and Raspberry Pi, and learn to build IoT solutions using modern communication protocols and APIs.

Course Outcome:

By the end of this course, students should be able to:

1. Understand the fundamental concepts, architecture, and applications of IoT.
2. Identify and utilize various hardware and software components involved in IoT systems.
3. Implement IoT solutions integrating sensors, actuators, and communication protocols.
4. Analyze data management strategies and security challenges in IoT.
5. Apply IoT knowledge to solve real-world problems in domains like healthcare, agriculture, and home automation.

Course Content/Syllabus:

Unit 1: Introduction to IoT

Architectural overview of IoT, Design principles and capabilities, IoT applications across industries, Introduction to sensing and actuation, Basics of networking in IoT, M2M communication, and IoT technology fundamentals.

Unit 2: Core IoT Technologies

Devices and gateways, Data management in IoT systems, Integration with business processes, Everything as a Service (XaaS), Role of cloud computing in IoT, Security aspects and best practices in IoT deployment.

Unit 3: IoT Hardware Components

Computing devices: Arduino, Raspberry Pi, Sensor and actuator technologies, Communication modules, I/O interfaces and their integration with hardware platforms.

Unit 4: IoT Software Components

Programming APIs using Python/Node.js/Arduino, Communication protocols: MQTT, ZigBee, Bluetooth, CoAP, UDP, TCP, Developing middleware for IoT data transmission and control.

Unit 5: Solution Frameworks and Case Studies

Frameworks for device integration, Data acquisition, Unstructured data storage (cloud/local), Authentication and authorization mechanisms, Case studies in industrial automation, transportation, agriculture, healthcare, and home automation.

Recommended Books:

1. Honbo Zhou, *The Internet of Things in the Cloud: A Middleware Perspective*, CRC Press
2. Vijay Madisetti, Arshdeep Bahga, *Internet of Things: A Hands-on Approach*, University Press
3. Pethuru Raj and Anupama C. Raman, *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*, CRC Press