Sant Gadge Baba Amravati University Amravati
Diploma in Applied Computer Technology (1 Year) [NEP-NSQF level 5]

Sr. No	type	Subject Code	Subject	Teaching Scheme (Hrs./Week)			Credit	Examination and Evaluation Scheme								
				Th.	Pr.	Total		Exam Duration (Hrs.)	Max. Marks				Total Marks		Min. Passing Marks	
									External Marks		Internal Marks			External	Internal	Total
									TH	PR	TH	PR				
	SEMESTER 1															
1	Core Skill	N1ACT1	Java Programming	3	-	3	3	3	50	-	30	-	80	20	12	32
2	Core Skill	N1ACT2	Linux Operating system	3	-	3	3	3	50	-	30	-	80	20	12	32
3	Core Skill	N1ACT3	Computer Network	3	-	3	3	3	50	-	30	-	80	20	12	32
4	Core Skill	N1ACT4	Web Technology	3	-	3	3	3	50	-	30	-	80	20	12	32
5	SEC/lab	N1ACT5	Java Lab	-	4	4	2	-	-	25	-	25	50	10	10	20
6	SEC/lab	N1ACT6	Operating system Lab	-	4	4	2	-	-	25	-	25	50	10	10	20
7	SEC/lab	N1ACT7	Web Technology Lab	-	4	4	2	-	-	25	-	25	50	10	10	20
8	GE	N1ACT8	Communication Skills	3	-	3	3	3	50	-	30	-	80	20	12	32
9	SEC/LAB	N1ACT9	Communication Skills-LAB	-	2	2	1	-	-	-	-	20	20	-	10	10
10	Assessment Hours					6										
			TOTAL			**35**	**22**						**570**			
	SEMESTER 2															
1	Core Skill	N2ACT1	Python Programming	3	-	3	3	3	50	-	30	-	80	20	12	32
2	Core Skill	N2ACT2	Database Management System	3	-	3	3	3	50	-	30	-	80	20	12	32
3	Core Skill	N2ACT3	Data Structures	3	-	3	3	3	50	-	30	-	80	20	12	32
4	DSE	N2ACTE1 N2ACTE2	E1- cloud computing E2- Data Mining & Warehousing	3	-	3	3	3	50	-	30	-	80	20	12	32
5	SEC	N2ACT4	LAB based on N2ACT E1/E2	-	4	4	2	-	-	25	-	25	50	10	10	20
6	SEC	N2ACT5	Lab based on N2ACT 1/2/3	-	4	4	2	-	-	25	-	25	50	10	10	20
7	SEC	N2ACT7	Computer hardware and Maintenance LAB	-	4	4	2	-	-	25	-	25	50	10	10	20
8	GE	N2ACT6	Computer hardware and Maintenance	2		2	2	2	50	-	30	-	80	20	12	32
9	SEC	N2ACT8	Project /Internship*		4	4	2	-	-	25	-	25	50	10	10	20
10	Assessment Hours					6										
			TOTAL			**36**	**22**						**600**			

*** Internship: Two Credits will be Awarded for minimum 60 Hrs of Internship**

Faculty: Science and Technology

Syllabus Prescribed for 1 Year Diploma in Applied Computer Technology UG Programme

[NEP-NSQF Level 5]

Programme Educational Objectives (PEO):

1. To equip students with foundational knowledge and practical skills in computer technology.
2. To provide students with hands-on experience in various aspects of applied computer technology.
3. To enable students to develop problem-solving abilities and critical thinking skills relevant to computer technology.
4. To prepare students for entry-level positions in the field of computer technology or further academic pursuits in related disciplines.

Programme Outcomes (PO's):

Upon successful completion of the diploma Programme, students should be able to:

1. Demonstrate understanding of fundamental concepts and principles in computer technology.
2. Apply various programming languages and tools to solve real-world problems.
3. Design and develop basic software applications and systems.
4. Analyze and troubleshoot common hardware and software issues.
5. Utilize computer networks and communication technologies effectively.
6. Work collaboratively in a team environment to accomplish computing tasks.
7. Communicate technical information effectively through oral, written, and visual means.
8. Adapt to emerging technologies and continue learning in the field of computer technology.

Programme Specific Outcomes (PSO's):

1. Apply computer hardware and software components effectively to solve specific technological challenges.
2. Demonstrate proficiency in utilizing specialized software tools relevant to applied computer technology.
3. Develop basic web applications to meet user requirements.
4. Configure and manage computer systems and networks for small-scale environments.
5. Evaluate and recommend appropriate computer solutions based on specific business or organizational needs.
6. Demonstrate awareness of ethical and legal considerations in the use of computer technology.
7. Collaborate with professionals from diverse backgrounds to implement interdisciplinary solutions involving computer technology.

EMPLOYBILITY:

1. **IT Support Technician:** As an IT support technician, you would provide technical assistance to computer system users. This could involve troubleshooting software and hardware issues, installing and configuring computer systems, and maintaining IT infrastructure.

2. **Computer Network Technician:** This role involves installing, maintaining, and troubleshooting computer networks within an organization. You would be responsible for ensuring network functionality, diagnosing and resolving network issues, and implementing network security measures.

3. **Database Administrator (DBA) Assistant:** Database administrators are responsible for managing and maintaining databases within an organization. As a DBA assistant, you would assist in tasks such as installing and configuring database software, performing database backups and recoveries, and ensuring data security.

4. **Web Developer:** With skills in web development, you could work on creating and maintaining websites and web applications for businesses and organizations. This could involve using programming languages such as HTML, CSS, JavaScript, and various web development frameworks.
5. **Software Tester/QA Analyst:** Quality assurance (QA) analysts are responsible for testing software applications to ensure they meet quality standards and perform as expected. You would design test cases, execute tests, and report and track defects to ensure the software meets user requirements.
6. **Technical Support Specialist:** Technical support specialists provide assistance to users experiencing technical issues with software, hardware, or other computer systems. This could involve troubleshooting problems, providing step-by-step instructions, and escalating complex issues to higher-level support teams.
7. **Computer Systems Analyst:** Computer systems analysts assess an organization's computer systems and procedures to help them operate more efficiently. This could involve analyzing user requirements, designing new systems or enhancements, and overseeing the implementation of technology solutions.
8. **IT Project Coordinator:** IT project coordinators assist in the planning, coordination, and execution of IT projects within an organization. This could involve tasks such as creating project plans, coordinating project resources, and monitoring project progress to ensure successful delivery.
9. **Desktop Support Specialist:** Desktop support specialists provide technical assistance and support to end-users for desktop hardware, software, and peripherals. This could involve installing and configuring desktop systems, troubleshooting issues, and providing training and guidance to users.

Semester-I

Title: Java Programming

Type: Core Skill

Credits: 3

Total Marks-80		Course Code: N1ACT1	(Total Number of Periods) Hrs
Theory Exam Marks :50	Internal Marks:30	Min Passing:32	45

Course Outcomes:

1. Develop and implement Java applications using object-oriented programming principles.
2. Utilize control flow statements to implement decision-making and looping logic.
3. Handle exceptions effectively in Java programs.
4. Manipulate arrays and strings efficiently.
5. Utilize collections framework for data storage and retrieval.
6. Implement file handling operations for input/output tasks.
7. Create multithreaded applications with synchronization.

Unit	Content
UNIT I:	**Fundamentals of Object-Oriented Programming:** Basic Concepts of Object-Oriented Programming (OOP), Benefits and Applications of OOP. **Java Evolution:** Java Features, Difference between Java, C and C++, Java and Internet, Java Environment. **Overview of Java Language:** Introduction to Simple Java Program, Use of Comments and Math function, Application of two classes, Java Program Structure, Java Tokens and statements, Implementing Java program and JVM, Command Line Arguments. **(11 Hours)**
UNIT II:	**Constants, Variables and Data Types:** Constants, Variables, Data Types, Declaration of Variables, Giving values to Variables, Symbolic Constants, Typecasting. **Operators & Expressions:** Arithmetic operators, Relational operators, Logical operators, Assignment operators, Increment & Decrement operators, conditional operators, Bitwise operators, Arithmetic Expressions, Evaluation of Expressions, Type Conversions in Expressions, Operator Precedence & Associativity. **Decision Making, Branching & Looping:** Decision Making with Control Statements, Looping statements, Jump in loops, Labelled loops. **(12 Hours)**
UNIT III:	**Classes, Objects and Methods:** Defining Class, Methods Declaration,Constructors, Methods Overloading, Overriding Methods, Inheritance **Arrays, Strings and Vectors:** 1D arrays, Creating an Array, 2D arrays, Strings,Vectors, Wrapper Classes, Enumerated Types **Inheritance:** Defining, extending classes, and Implementing Interfaces. Multipleinheritance and polymorphism. **(11 Hours)**
UNIT IV:	**Packages:** Basics of packages, System packages, Creating and accessing packages, Creating user defined packages, Adding class to a package. **Exception Handling:** Using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, Multiple catch statements, Creating user defined exceptions.**(11 Hours)**

Reference Book:

1 Junaid Khateel & Dr. G. T.Thampi Computer Programming in JAVA DreamTech Press
2 Sharnam Shah & Vaishali Shah Core JAVA for Beginners SPD
3 E Balagurusamy Programming in JAVA a primer TMH
4 Sachin Malhotra & Saurabh Chaudhary Programming in JAVA Oxford University Press
5 Rashmi Kanta Das Core Java for beginners Vikas Publishing House Pvt. Ltd

Title: Operating system

Type: Core Skill

Credits: 3

Total Marks-80		Course Code: N1ACT2	(Total Number of Periods) Hrs
Theory Exam Marks :50	Internal Marks:30	Min Passing:32	45

Course Outcomes:

1. Fundamental understanding of the role of Operating Systems.
2. To understand the concept of a process and thread.
3. To understand the various memory management techniques.
4. To apply the cons of process/thread scheduling.
5. To apply the concept of process synchronization, mutual exclusion and the deadlock.
6. To realize the concept of I/O management and File system.

UNIT – I OVERVIEW OF OPERATING SYSTEM (11 Hours)

Operating System Objectives and Functions, The Evolution of Operating Systems, Developments Leading to Modern Operating Systems, Virtual Machines. BASH Shell scripting: Basic shell commands, shell as a scripting language.

UNIT – II PROCESS DESCRIPTION AND CONTROL (11 Hours)

Process: Concept of a Process, Process States, Process Description, Process Control (Process creation, Waiting for the process/processes, Loading programs into processes and Process Termination), Execution of the Operating System.

Threads: Processes and Threads, Concept of Multithreading, Types of Threads, Thread programming Using Pthreads. Scheduling: Types of Scheduling, Scheduling Algorithms, and Thread Scheduling.

UNIT – III CONCURRENCY CONTROL (11 Hours)

Process/thread Synchronization and Mutual Exclusion: Principles of Concurrency, Requirements for Mutual Exclusion, Mutual Exclusion: Hardware Support, Operating System Support (Semaphores and Mutex), Programming Language Support (Monitors). Classical synchronization problems: Readers/Writers Problem, Producer and Consumer problem, Interprocess communication Deadlock: Principles of Deadlock, Deadlock Modeling, Deadlock Prevention, Deadlock Avoidance, Deadlock detection and recovery.

UNIT – IV MEMORY, I/O & FILE MANAGEMENT (12 Hours)

Memory Management: Memory Management Requirements, Memory Partitioning, Buddy System, Relocation, Paging, Segmentation. Virtual Memory: Hardware and Control Structures, Operating System Software. I/O Management and Disk Scheduling: I/O Devices, Organization of the I/O Function, Operating System Design Issues, I/O Buffering, Disk Scheduling, Disk Cache. File Management: Overview, File Organization and Access, File Directories, File Sharing, Record Blocking, Secondary Storage Management.

Reference Book:

1. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, ISBN-10: 0-13-380591-3, ISBN-13: 978-0-13-380591-8, 8th Edition
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, WILEY, ISBN 978-1-118-06333-0 , 9th Edition
3. Andrew S. Tanenbaum & Herbert Bos, Modern Operating System, Pearson, ISBN-13: 9780133592221, 4th Edition
4. "Linux Bible" by Christopher Negus
5. "How Linux Works: What Every Superuser Should Know" by Brian Ward

Title: Computer Network

Type: Core Skill

Credits: 3

Total Marks-80		Course Code: N1ACT3	(Total Number of Periods) Hrs
Theory Exam Marks :50	Internal Marks:30	Min Passing:32	45

Course Outcomes:

1. Gain the knowledge of the basic computer network technology.
2. Gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.

UNIT – I (11 Hours)

Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet. Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless Transmission. Data link layer: Design issues, framing, Error detection and correction.

UNIT - II (11 Hours)

Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channels. Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols. Medium Access sublayer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

UNIT - III (11 Hours)

Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.

UNIT - IV (12 Hours)

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols. Application Layer –Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video

Reference Book:

1. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI
2. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education
3. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH

Title: Web Technology

Type: Core Skill

Credits: 3

Total Marks-80		Course Code: N1ACT4	(Total Number of Periods) Hrs
Theory Exam Marks :50	Internal Marks:30	Min Passing:32	45

Course Outcomes:

1. Implement and analyze behavior of web pages using HTML and CSS
2. Apply the client side technologies for web development
3. Analyze the concepts of Servlet and JSP
4. Analyze the Web services and frameworks
5. Apply the server side technologies for web development
6. Create the effective web applications for business functionalities using latest web development platforms

Unit 1: Introduction to Web Development (11 Hours)

The Internet, basic internet protocols, the world wide web, HTTP Request message, HTTP response message, web clients, web servers. HTML: Introduction, history and versions. HTML elements: headings, paragraphs, line break, colors and fonts, links, frames, lists, tables, images and forms, Difference between HTML and HTML5. CSS: Introduction to Style Sheet, CSS features, CSS core syntax, Style sheets and HTML, Style rule cascading and inheritance, text properties. Bootstrap.

Unit 2: Client Side Technologies (11 Hours)

JavaScript: Introduction to JavaScript, JavaScript in perspective, basic syntax, variables and data types, statements, operators, literals, functions, objects, arrays, built in objects, JavaScript debuggers.

DOM: Introduction to Document Object Model, DOM history and levels, intrinsic event handling, modifying element style, the document tree, DOM event handling, jQuery, Overview of Angular JS.

Unit 3: Java Servlets and XML (11 Hours)

Servlet: Servlet architecture overview, A “Hello World” servlet, Servlets generating dynamic content, Servlet life cycle, parameter data, sessions, cookies, URL rewriting, other Servlet capabilities, data storage, Servlets concurrency, databases (MySQL) and Java Servlets. XML: XML documents and vocabularies, XML declaration, XML Namespaces, DOM based XML processing, transforming XML documents, DTD: Schema, elements, attributes. AJAX: Introduction, Working of AJAX.

Unit 4:JSP & Web Services (12 Hours)

JSP: Introduction to Java Server Pages, JSP and Servlets, running JSP applications, Basic JSP, JavaBeans classes and JSP, Support for the Model-view-controller paradigm, JSP related technologies.

Web Services: Web Service concepts, Writing a Java Web Service, Writing a Java web service client, Describing Web Services: WSDL, Communicating Object data: SOAP. Struts: Overview, architecture, configuration, actions, interceptors, result types, validations, localization, exception handling, annotations

Reference Book:

1. Jeffrey C.Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035
2. Robert W. Sebesta, “Programming the World Wide Web”, 4th Edition, Pearson education, 2008.
3. "HTML and CSS: Design and Build Websites" by Jon Duckett
4. "JavaScript and JQuery: Interactive Front-End Web Development" by Jon Duckett
5. "Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5" by Robin Nixon.

Title: Java Lab

Type: SEC/lab

Credits: 2

Total Marks-50		Course Code: N1ACT5	(Total Number of Periods) Hrs
Theory Exam Marks :25	Internal Marks:25	Min Passing:20	60

Course Title: Java – LAB

List of Practical’s:

NOTE: The list suggests sample program set. Hence, the concerned staff may modify the list as needed (Minimum 15).

1. Write a Java program to print "Hello, World!" to the console.
2. Create a program to take user input for their name and age, then print a personalized greeting message.
3. Write a program to calculate and print the area of a rectangle given its length and width.
4. Implement a program that checks if a given number is even or odd and prints the result.
5. Create a Java program to generate a multiplication table for a given number.
6. Define a class representing a Car with attributes like make, model, and year. Create objects of this class and print their details.
7. Write a program to demonstrate method overloading by creating multiple methods with the same name but different parameters to calculate the area of a shape (e.g., rectangle, circle).
8. Create a class representing a BankAccount with methods to deposit, withdraw, and check balance. Instantiate objects of this class and perform transactions.
9. Implement a program to illustrate method overriding by creating a superclass Animal and subclasses like Dog and Cat with a common method sound().
10. Design a class representing a Student with attributes like name, roll number, and marks. Write methods to calculate the total and average marks of students.
11. Define an interface named Shape with methods to calculate area and perimeter. Implement this interface in classes like Circle and Rectangle.
12. Create a package named "math" and define classes for basic mathematical operations like addition, subtraction, multiplication, and division.
13. Design an interface named Vehicle with methods to start(), stop(), and accelerate(). Implement this interface in classes like Car and Bicycle.

14. Implement a package named "utilities" containing classes for string manipulation operations like reversing a string and checking for palindrome.
15. Define an interface named Animal with methods sound() and eat(). Implement this interface in classes like Dog and Cat.
16. Write a program to handle ArithmeticException by dividing a number by zero within a try-catch block.
17. Create a custom exception class called "InvalidAgeException" to handle invalid age inputs.
18. Implement a multithreaded program to print even and odd numbers alternatively using two threads.
19. Write a program to demonstrate synchronization by implementing a shared resource accessed by multiple threads.
20. Design a program to simulate a bank ATM with withdrawal and deposit operations handled by multiple threads.
21. Create a Java applet to draw simple shapes like circles and rectangles.
22. Write a program to read data from a text file and display it on the console.
23. Implement a program to store student records using ArrayList and perform operations like add, delete, and display records.
24. Design a program to sort an array of integers using different sorting algorithms like bubble sort or quicksort.
25. Write a program to serialize and deserialize a collection of objects (e.g., student records) to/from a file.

Title: Operating system Lab

Type: SEC/lab

Credits: 2

Total Marks-50		Course Code: N1ACT6	(Total Number of Periods) Hrs
External Marks:25	Internal Marks:25	Min Passing:20	60

List of Practical's:

NOTE: The list suggests sample program set. Hence, the concerned staff is may modify the list as needed (Minimum 15).

1. Write Java code to simulate the following CPU Scheduling algorithms
   a) FCFS b) SJF c) Round Robin d) priority
2. Write programs using the I/O system calls of LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir)
3. Write a Java code to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
4. using LINUX system calls implement the Producer – Consumer problem using semaphores.
5. Write Java code to illustrate the following IPC mechanisms a) Pipes b) FIFOs c) Message Queues d) Shared Memory
6. Write Java code to simulate the following memory management techniques a) Paging b) Segmentation
7. Write Java code to simulate Page replacement policies a) FCFS b) LRU c) Optimal.
8. Install a Linux distribution (e.g., Ubuntu, Fedora, CentOS) on a virtual machine or as a dual boot.
9. Execute basic linux commands such as **ls**, **cd**, **pwd**, **mkdir**, **rm**, **cp**, **mv**, **cat**, **nano** or **vi** for text editing.
10. **File System Navigation:**

a. Practice navigating through the file system using commands like **cd**, **ls**, **pwd**, and **tree**.

b. Learn about the Linux file system hierarchy (/bin, /etc, /home, /var, etc.).

11. **User and Group Management:**

a. Create users and groups using **useradd**, **groupadd**, **usermod**, **groupmod**.

b. Assign permissions using **chmod** and **chown**.

c. Explore **/etc/passwd**, **/etc/group**, **/etc/shadow** files.

12. **Package Management:**

a) Use package managers like **apt** (Debian/Ubuntu), **yum** or **dnf** (Fedora/CentOS), **pacman** (Arch Linux), or **zypper** (openSUSE) to install, update, and remove software packages.

b) Learn basic package management commands such as **apt install**, **apt update**, **apt upgrade**, **apt remove**.

13. **Process Management:**

a) Use commands like **ps**, **top**, **kill**, **killall**, **pgrep**, and **pkill** to manage processes.

b) Practice backgrounding and foregrounding processes.

c) Learn about process priorities with **nice** and **renice**.

14. **File Permissions and Ownership:**

a) Understand Linux file permissions (**chmod**) and ownership (**chown**, **chgrp**).

b) Practice changing permissions and ownership on files and directories.

15. **Networking:**

a) Learn basic networking commands such as **ping**, **ifconfig** (or **ip**), **netstat**, **traceroute**, **nslookup**, **dig**.

b) Configure network interfaces (**ifconfig**, **ip**).

c) Explore **/etc/hosts**, **/etc/hostname**, **/etc/network/interfaces**.

16. **System Monitoring:**

a) Use tools like **htop**, **iotop**, **iftop**, and **nload** to monitor system resources (CPU, memory, disk, network).

b) Explore **/proc** directory for system information.

Title: Web Technology Lab

Type: SEC/LAB

Credits: 2

Total Marks-50		Course Code: N1ACT7	(Total Number of Periods) Hrs
External Marks:50	Internal Marks:00	Min Passing:20	30

List of Practical's:

NOTE: The list suggests sample program set. Hence, the concerned staff is may modify the list as needed (Minimum 15).

1. **Case Study: Website Evaluation**
    - Visit at least 5 different websites.
    - Note down the URL, purpose, things liked, things disliked, and overall evaluation for each website.
    - Analyze the evaluations to identify common design issues and considerations for website development.
2. **Implement a Web Page using HTML**
    - Create an **index.htm** file for a client website (e.g., restaurant) using HTML.
    - Utilize HTML tags such as headings, paragraphs, frames, tables, images, lists, links, and forms.
    - Experiment with different attributes of HTML tags.
    - Incorporate both internal CSS, inline CSS, and external CSS for styling the web page.
3. **Design XML Document with DTD and XML Schema**
    - Develop an XML document structure to store employee information for a business organization.
    - Create a DTD (Document Type Definition) and an XML Schema to validate the XML document.
    - Apply CSS/XSL to display the content of the XML document, possibly in a tabular format.
4. **Implement an Application in JavaScript**
    - Design a user interface using HTML and CSS for the application (e.g., a simple calculator).
    - Write JavaScript code to implement functionality such as input validation, calculations, and interaction with the user.
    - Utilize prompt and alert windows for user input and feedback.
5. **Implement a Servlet Program**
    - Create a database table (e.g., **ebookshop**) using Oracle/MySQL.
    - Develop a Servlet to interact with the database (e.g., retrieve and display table content using SQL select query).
6. **Implement a JSP Program**
    - Similar to the Servlet implementation, create a database table (e.g., **students_info**) and populate it.
    - Use JSP to retrieve and display the table content using SQL select query.
7. **Build a Dynamic Web Application using PHP and MySQL**
    - Design database tables in MySQL.
    - Establish a connection between PHP and MySQL.
    - Implement add, update, delete, and retrieve functions in PHP for interacting with the MySQL database.

8. **Design a Login Page with Struts**
    - Create a login page using Struts framework.
    - Implement validations for name, mobile number, email id, and empty fields.
    - Display error messages for invalid inputs and a welcome page upon successful login.
9. **Design an Application using AngularJS**
    - Develop registration and login pages using AngularJS.
    - Implement form validation and user authentication features.
10. **Implement Business Logic using EJB**
    - Design and implement a web application logic for deposit and withdraw transactions using EJB.
    - Ensure proper transaction handling and security measures.

Title: Communication Skills - I

Type: GE

Credits: 3

Total Marks-80	Course Code: N1ACT8	(Total Number of Periods) Hrs: 45(T)
Theory Exam Marks :50	Theory Internal Marks:30	Min Passing:32

COURSE OUTCOME:

At end of the course students would be able to :

1. understand communication skills of English language
2. Formulate/ compose his own sentences and able to speak English Language.
3. collaborate with others students in English.
4. communicate properly their ideas and concepts in English.

Unit	Content
Unit 1:	o Articles o Prepositions o Tenses o Subject – Verb Agreement **(11 Hours)**
Unit 2:	o Meeting People o Exchanging Greetings and Taking Leave o Introducing Yourself **(11 Hours)**
Unit 3: Prose	o The Home Coming – Rabindranath Tagore o A Lesson My Father Taught Me – APJ Abdul Kalam o How I Became a Public Speaker – George Bernard Shaw **(11 Hours)**
Unit 4: Poetry	o The quality of Mercy – William Shakespeare o The Mountain and the Squirrel – R.W. Emerson

	o Where the Mind is Without Fear – Rabindranath Tagore **(12 Hours)**

TEXT BOOK:-

Pathmaker: A Textbook for College Students [ISBN 989354421778] Edited by Board of Editors, SantGadge Baba Amravati University, Amravati. Publisher : Orient BlackSwan Pvt L

Title: Communication Skills-LAB

Type: SEC/LAB

Credits: 1

Total Marks-20	Course Code: N1ACT9	(Total Number of Periods) Hrs: 30
Theory Internal Marks:20		Min Passing:10

NOTE: The list suggests sample program set. Hence, the concerned staff may modify the list as needed (Minimum 10).

LIST OF PRACTICAL

1 Letter writing formal and informal

2 Email

3 Resume

4 Making Request

5 Responding to thanks

6 Blog writing

7 Application writing

8 Question tags

9 Zero suffix and infix

10 Requesting

SEMESTER II

Title: Python Programming

Type: Core Skill

Credits: 3

Total Marks-80		Course Code: N2ACT1	(Total Number of Periods) Hrs
External Marks:50	Internal Marks:30	Min Passing:32	45

Course Outcomes:

1. Develop algorithmic solutions to simple computational problems.
2. Develop and execute simple Python programs.
3. Decompose a Python program into functions.
4. Represent compound data using Python lists, tuples, dictionaries etc.
5. Read and write data from/to files in Python programs.
6. Write simple Python programs using conditionals and loops for solving problems.

UNIT I: COMPUTATIONAL THINKING AND PROBLEM SOLVING (11 Hours)

Fundamentals of Computing – Identification of Computational Problems -Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion). Illustrative problems: find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.

UNIT II: DATA TYPES, EXPRESSIONS, STATEMENTS (11 Hours)

Python interpreter and interactive mode,debugging; values and types: int, float, boolean, string , and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; Illustrative programs: exchange the values of two variables, circulate the values of n variables, distance between two points.

UNIT III: CONTROL FLOW, FUNCTIONS, STRINGS (11 Hours)

Conditionals:Boolean values and operators, conditional (if), alternative (if-else),chained conditional (if-elif-else);Iteration: state, while, for, break, continue, pass; Fruitful functions: return values,parameters, local and global scope, function composition, recursion; Strings: string slices,immutability, string functions and methods, string module; Lists as arrays. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.

UNIT IV: LISTS, TUPLES, DICTIONARIES AND FILES (12 Hours)

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: simple sorting, histogram, Students marks statement, Retail bill preparation.

Files and exceptions: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules

Reference Books:

1. Allen B. Downey, “Think Python: How to Think like a Computer Scientist”, 2nd Edition, O’Reilly Publishers, 2016
2. Karl Beecher, “Computational Thinking: A Beginner's Guide to Problem Solving and Programming”, 1st Edition, BCS Learning & Development Limited, 2017.
3. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021.
4. G Venkatesh and Madhavan Mukund, “Computational Thinking: A Primer for Programmers and Data Scientists”, 1st Edition, Notion Press, 2021.

Title: Database Management System

Type: Core Skill

Credits: 3

Total Marks-80		Course Code: N2ACT2	(Total Number of Periods) Hrs
External Marks:50	Internal Marks:30	Min Passing:32	45

Course Outcomes:

1. Model application data requirements and design database schemas.
2. Understand and address security issues in DBMS.
3. Formulate solutions to query problems using relational algebra/SQL.
4. Demonstrate understanding of normalization theory and apply it to database normalization.
5. Run transactions and control consequences of concurrent data access.

UNIT I: RELATIONAL DATABASES (11 Hours)

Data Models, Relational Data Models, Relational Algebra, Structured Query Language (SQL), Entity-Relationship Model (ER Model), Mapping ER Models to Relations, Distributed Databases, Data Fragmentation, Replication

UNIT II: DATABASE DESIGN (11 Hours)

ER Diagrams, Functional Dependencies, Non-Loss Decomposition Functional Dependencies, First Normal Form, Second Normal Form, Third Normal Form, Dependency Preservation ,Boyce/Codd Normal Form, Multi-Valued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form

UNIT III: TRANSACTION MANAGEMENT (11 Hours)

Transaction Concepts, ACID Properties, Serializability, Transaction Isolation Levels, Concurrency Control, Need for Concurrency, Lock-Based Protocols, Deadlock Handling, Recovery System, Failure Classification, Recovery Algorithm

UNIT IV: DATABASE SECURITY (12 Hours)

Need for database security, SQL Injection Attacks, The Injection Technique, SQLi Attack Avenues and Types

Reference Books:

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Seventh Edition, Tata McGraw Hill, 2021.
2. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Seventh Edition, Pearson Education, 2016.
3. William Stallings, Lawrie Brown, “Computer Security: Principles and Practice”, Fourth Edition, Pearson, 2019.
4. C.J. Date, A. Kannan, and S. Swamynathan, “An Introduction to Database Systems”, Pearson Education, Eighth Edition, 2006.
5. Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems”, Third Edition, McGraw Hill, 2014.

Title: Data Structures

Type: Core Skill

Credits: 3

Total Marks-80		Course Code: N2ACT3	(Total Number of Periods) Hrs
External Marks:50	Internal Marks:30	Min Passing:32	45

Course Outcomes:

1. Ability to select the data structures that efficiently model the information in a problem.
2. Ability to assess efficiency trade-offs among different data structure implementations or combinations.
3. Implement and understand the application of algorithms for sorting and pattern matching.
4. Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and AVL-trees.

UNIT I: Introduction to Data Structures (11 Hours)

Abstract data types, Linear list – singly linked list implementation, Insertion, deletion, and searching operations on linear list, Stacks: Operations, array and linked representations, stack applications, Queues: Operations, array and linked representations

UNIT II: Dictionaries and Hash Table Representation (11 Hours)

Linear list representation, Skip list representation, Operations: insertion, deletion, and searching, Hash Table Representation: hash functions, collision resolution (separate chaining, open addressing - linear probing, quadratic probing, double hashing), rehashing, extendible hashing

UNIT III: Search Trees (11 Hours)

Binary Search Trees: Definition, Implementation, Operations (Searching, Insertion, and Deletion), B-Trees, B+ Trees, AVL Trees: Definition, Height of an AVL Tree, Operations (Insertion, Deletion, and Searching), Red-Black Trees, Splay Trees

UNIT IV: Graphs and Sorting (12 Hours)

Graph Implementation Methods, Graph Traversal Methods, Sorting: Quick Sort, Heap Sort, External Sorting (Model for external sorting, Merge Sort)

Reference Book:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni, and Susan Anderson Freed, Universities Press.
2. Data Structures using C – A. S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.
3. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B.A.Forouzan, Cengage Learning.

Title: cloud computing

Type: DSE

Credits: 3

Total Marks-80		Course Code: N2ACTE1	(Total Number of Periods) Hrs
External Marks:50	Internal Marks:30	Min Passing:32	45

Course Outcomes:

1. To install cloud computing environments.
2. To develop any one type of cloud

3. To explore future trends of cloud computing

Unit I: Basics of Cloud Computing (11 Hours)

Overview, Applications, Intranets and the Cloud. Your Organization and Cloud Computing- Benefits, Limitations, Security Concerns. Software as a Service (SaaS)- Understanding the Multitenant Nature of SaaS Solutions, Understanding SOA. Platform as a Service (PaaS)-IT Evolution Leading to the Cloud, Benefits of Paas Solutions, Disadvantages of PaaS Solutions. Infrastructure as a Service (IaaS)-Understanding IaaS, Improving Performance through Load Balancing, System and Storage Redundancy, Utilizing Cloud-Based NAS Devices, Advantages, Server Types. Identity as a Service (IDaaS).

Unit II: Data Storage and Security in Cloud (11 Hours)

Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo Cloud data stores: Datastore and Simple DB Gautam Shrauf, Cloud Storage-Overview, Cloud Storage Providers. [Anthony T. Velte]3 Securing the Cloud- General Security Advantages of Cloud-Based Solutions, Introducing Business Continuity and Disaster Recovery. Disaster Recovery- Understanding the Threats.

Unit III Virtualization (12 Hours)

Implementation Levels of Virtualization, Virtualization Structures/Tools and Mechanisms, Types of Hypervisors, Virtualization of CPU, Memory, and I/O Devices, Virtual Clusters and Resource Management, Virtualization for Data-Center Automation. Common Standards: The Open Cloud Consortium, Open Virtualization Format, Standards for Application Developers: Browsers (Ajax), Data (XML, JSON), Solution Stacks (LAMP and LAPP),Syndication (Atom, Atom Publishing Protocol, and RSS), Standards for Security.

Unit IV: Ubiquitous Clouds and the Internet of Things (11 Hours)

Cloud Trends in Supporting Ubiquitous Computing, Performance of Distributed Systems and the Cloud, Enabling Technologies for the Internet of Things (RFID, Sensor Networks and ZigBee Technology, GPS), Innovative Applications of the Internet of Things (Smart Buildings and Smart Power Grid, Retailing and Supply-Chain Management, Cyber-Physical System), Online Social and Professional Networking

Reference Books:

1. Anthony T. Velte Toby J. Velte, Robert Elsenpeter, “Cloud Computing: A Practical Approach”, 2010, The McGraw-Hill.
2. Dr. Kris Jamsa, “ Cloud Computing: SaaS, PaaS, IaaS, Virtualization and more” , Wiley Publications, ISBN: 978-0-470-97389-9
3. Gautam Shrof, “ENTERPRISE CLOUD COMPUTING Technology Architecture, Applications, Cambridge University Press, ISBN: 9780511778476 References

Title: E2- Data Mining & Warehousing

Type: DSE

Credits: 3

Total Marks-80		Course Code: N2ACTE2	(Total Number of Periods) Hrs
External Marks:50	Internal Marks:30	Min Passing:32	45

Course Outcomes:

1. Apply basic, intermediate and advanced techniques to mine the data
2. Analyze the output generated by the process of data mining
3. Explore the hidden patterns in the data
4. Optimize the mining process by choosing best data mining technique

Unit I: Introduction (11 Hours)

Data Mining, Data Mining Task Primitives, Data: Data, Information and Knowledge; Attribute Types: Nominal, Binary, Ordinal and Numeric attributes, Discrete versus Continuous Attributes; Introduction to Data

Preprocessing, Data Cleaning: Missing values, Noisy data; Data integration: Correlation analysis; transformation: Min-max normalization, z-score normalization and decimal scaling; data reduction: Data Cube Aggregation, Attribute Subset Selection, sampling; and Data Discretization: Binning, Histogram Analysis

Unit II: Data Warehouse (11 Hours)

Data Warehouse, Operational Database Systems and Data Warehouses(OLTP Vs OLAP), A Multidimensional Data Model: Data Cubes, Stars, Snowflakes, and Fact Constellations Schemas; OLAP Operations in the Multidimensional Data Model, Concept Hierarchies, Data Warehouse Architecture, The Process of Data Warehouse Design, A three-tier data warehousing architecture, Types of OLAP Servers: ROLAP versus MOLAP versus HOLAP.

Unit III: Measuring Data Similarity and Dissimilarity (11 Hours)

Measuring Data Similarity and Dissimilarity, Proximity Measures for Nominal Attributes and Binary Attributes, interval scaled; Dissimilarity of Numeric Data: Minskowski Distance, Euclidean distance and Manhattan distance; Proximity Measures for Categorical, Ordinal Attributes, Ratio scaled variables; Dissimilarity for Attributes of Mixed Types, Cosine Similarity.

Unit IV: Association Rules Mining (12 Hours)

Market basket Analysis, Frequent item set, Closed item set, Association Rules, a-priori Algorithm, Generating Association Rules from Frequent Item sets, Improving the Efficiency of a-priori, Mining Frequent Item sets without Candidate Generation: FP Growth Algorithm; Mining Various Kinds of Association Rules: Mining multilevel association rules, constraint based association rule mining, Meta rule-Guided Mining of Association Rules.

Reference Books:

1. Han, Jiawei Kamber, Micheline Pei and Jian, “Data Mining: Concepts and Techniques”, Elsevier Publishers, ISBN:9780123814791, 9780123814807.
2. Parag Kulkarni, “Reinforcement and Systemic Machine Learning for Decision Making” by Wiley-IEEE Press, ISBN: 978-0-470-91999-6
3. Matthew A. Russell, "Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More" , Shroff Publishers, 2nd Edition, ISBN: 9780596006068
4. Maksim Tsvetovat, Alexander Kouznetsov, "Social Network Analysis for Startups: Finding connections on the social web", Shroff Publishers , ISBN: 10: 1449306462

Title: LAB based on N2ACT E1/E2

Type: SEC

Credits: 2

Total Marks-50		Course Code: N2ACT4	(Total Number of Periods) Hrs
External Marks:25	Internal Marks:25	Min Passing:20	60

Minimum 15 experiments / programming assignments must be completed based on the respective syllabus (N2ACT E1/ N2ACT E2).

Title: LAB based on N2ACT 1/2/3

Type: SEC

Credits: 2

Total Marks-50		Course Code: N2ACT5	(Total Number of Periods) Hrs
External Marks:25	Internal Marks:25	Min Passing:20	60

Minimum 15 experiments / programming assignments must be completed based on the respective syllabus (N2ACT1, N2ACT2, N2ACT3).

Title: Computer hardware and Maintenance

Type: GE

Credits: 3

Total Marks-80	Course Code: N2ACT6	(Total Number of Periods) Hrs:45(T)
Theory Exam Marks :50	Theory Internal Marks:30	Min Passing:32

Course Outcomes:

1. Understand the fundamental principles of computer hardware architecture.
2. Identify and describe the function of various hardware components within a computer system.
3. Demonstrate proficiency in assembling and disassembling computer hardware.
4. Implement troubleshooting techniques to diagnose and resolve hardware issues.
5. Develop preventive maintenance strategies to prolong the lifespan of computer systems.

Unit 1: Introduction to Computer Hardware (11 Hours)

- Overview of computer hardware components
- Understanding computer architecture and organization
- Role of CPU, RAM, motherboard, and power supply unit
- Introduction to input/output devices and storage devices

Unit 2: Central Processing Unit (CPU) and Motherboard (11 Hours)

- Functions and types of CPUs
- Motherboard components and their roles
- BIOS/UEFI setup and configuration
- CPU installation, cooling systems, and thermal management

Unit 3: Memory and Storage Devices (11 Hours)

- Types, working and functions of memory (RAM, ROM, Cache)
- Hard disk drives (HDDs) vs. Solid State Drives (SSDs)
- Optical drives and their usage
- Memory and storage expansion options

Unit 4: Peripheral Devices and Connectivity (12 Hours)

- Input devices (keyboard, mouse, scanner, etc.) and their configurations
- Output devices (monitor, printer, speakers, etc.) and connections
- Understanding ports and interfaces (USB, HDMI, Ethernet, etc.)
- Network devices and configurations (routers, switches, modems)

Reference Books:

1. "CompTIA A+ Certification All-in-One Exam Guide, Tenth Edition" by Mike Meyers

2. "Upgrading and Repairing PCs" by Scott Mueller
3. "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy
4. "IT Essentials: PC Hardware and Software Companion Guide" by Cisco Networking Academy

Title: COMPUTER HARDWARE AND MAINTAINENCE - LAB

Type: SEC

Credits: 2

Total Marks-50		Course Code: N2ACT7	(Total Number of Periods) Hrs
External Marks:25	Internal Marks:25	Min Passing:20	60

- Minimum 15 experiments / programming assignments must be completed based on the respective syllabus

Practical Exercises:

1. Understanding Basic Components:

- **Processor (CPU)**: Learn about different types of processors, their specifications, and how to install and remove them.
- **Motherboard**: Understand the layout, different ports, and how to connect components to it.
- **RAM**: Learn about different types of RAM, how to install and remove RAM modules.
- **Storage (HDD/SSD)**: Understand the difference between HDD and SSD, how to install and manage them.
- **Power Supply Unit (PSU)**: Learn about power requirements, cable management, and how to replace a PSU.
- **Graphics Card (GPU)**: Understand its role, types, and how to install and replace a GPU.

2. Building a PC:

- Start with assembling a basic PC. Follow guides available online or in manuals provided with components.
- Practice cable management to ensure good airflow and aesthetics.
- Install the operating system (OS) once the hardware is assembled.

3. Troubleshooting:

- Learn common hardware issues and their solutions.
- Practice diagnosing problems like overheating, hardware conflicts, and component failures.
- Understand BIOS/UEFI settings for troubleshooting and optimizing hardware performance.

4. Maintenance:

- Regularly clean dust from fans, heatsinks, and other components using compressed air.
- Ensure proper ventilation and cooling to prevent overheating.
- Update drivers and firmware regularly for optimal performance and security.
- Implement a backup strategy for important data.

5. Advanced Skills:

- Learn about overclocking CPU/GPU for improved performance (if applicable).
- Practice upgrading components such as RAM, storage, or graphics cards.
- Explore more about water cooling systems (if interested).

6. Safety Measures:

- Always disconnect the power source before working on hardware.
- Handle components with care, avoiding static electricity and physical damage.
- Use appropriate tools for the job and follow manufacturer guidelines.

PROJECT WORK:

Total Marks-50	Course Code: N2ACT8		(Total Number of Periods) Hrs:60
CREDIT:2	External Marks:25	Internal:25	Min Passing:20

Students pursuing a Diploma in Advanced Computer Technology are required to undertake a project that demonstrates their understanding and application of cybersecurity concepts, techniques, and tools. The project work serves as a culmination of their learning experience and allows them to showcase their skills in a practical setting. Below are some footnotes regarding the project work for the syllabus:

1. **Project Proposal Submission:** Students are required to submit a project proposal outlining the scope, objectives, methodology, and expected outcomes of their project. The proposal should be reviewed and approved by the faculty before proceeding with the project.
2. **Project Selection:** Students have the flexibility to choose a project topic within the domain of cybersecurity based on their interests and career aspirations. The project could focus on areas such as network security, cryptography, digital forensics, incident response, or ethical hacking.
3. **Project Execution:** Students are expected to demonstrate proficiency in planning, executing, and documenting their project work. This involves conducting research, implementing appropriate methodologies and techniques, and adhering to best practices in cybersecurity.
4. **Hands-on Implementation:** The project should incorporate hands-on implementation, where students apply theoretical concepts learned in the classroom to real-world scenarios. This may involve setting up a lab environment, performing experiments, conducting security assessments, or developing security solutions.
5. **Documentation and Reporting:** Students are required to maintain detailed documentation throughout the project, including design documents, implementation logs, test results, and analysis findings. A final project report summarizing the entire project lifecycle, including methodology, findings, challenges, and recommendations, should be submitted.
6. **Presentation and Defense:** Upon completion of the project, students are expected to deliver a presentation to the faculty and peers, highlighting the key aspects of their project. They should be prepared to answer questions and defend their methodology, findings, and conclusions.
7. **Evaluation Criteria:** The project work will be evaluated based on various criteria, including the relevance of the topic, technical depth, creativity, quality of implementation, documentation clarity, presentation skills, and overall contribution to the field of cybersecurity.
8. **Ethical Considerations:** Students must adhere to ethical guidelines and principles throughout the project work, ensuring that their activities do not violate privacy, integrity, or confidentiality laws and regulations. Any ethical concerns or potential risks should be addressed and mitigated appropriately.

*Internship:

Internship will be conducted after Ist semester in vacations for minimum 60 hrs. It's 2 credits will be reflected in final semester credit grade report.