



SEVA MANDAL EDUCATION SOCIETY'S
Dr. BHANUBEN MAHENDRA NANAVATI COLLEGE OF HOME SCIENCE
(Empowered Autonomous)
NAAC Re-accredited 'A+' Grade with CGPA 3.69/4 (3rd Cycle)
UGC Status: College with Potential for Excellence
Selected under "Enhancing Quality and Excellence in Select Autonomous College"
by Rashtriya Uchchatar Shiksha Abhiyan (RUSA)

STRUCTURE AND SYLLABUS OF SEMESTER I
(NEP 2020 GUIDELINES)

ACADEMIC YEAR-2025-2026

DEPARTMENT OF COMPUTER APPLICATIONS

STRUCTURE OF SEMESTER I

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY (B.SC. IT)

SN	Course Code	Courses	Type of Course	Credits	Theory	Practical /Tutorials	Marks
1.1	041UITMJ011	Imperative Programming	Major (Core)	4	4	-	100
1.2	041UITMJ012	Operating System	Major (Core)	2	2	-	50
1.3	041UITOE013	Digital Electronics and Logic Circuits	OEC	4	4	-	100
1.4	041UITVS014	Imperative Programming Lab	VSC	2	1	1	50
1.5	041UITSE015	GNU/ Linux programming Lab	SEC	2	1	1	50
1.6	041UITAE016	Proficiency in English Communication	AEC	2	1	1	50
1.7	041UITIK017	Living Heritage of India	IKS	2	1	1	50
1.8	041UITVE018A / 041UITVE018B	Digital and Technological Solutions (Accounting Applications / Office Automation)	VEC	2	1	1	50
1.9	041UITCC019A / 041UITCC019B	NSS/ Health and Wellness	CC	2	1	1	50
Total				22			550

1.1 Major (Core)

Course Code	Course	TC	Th C	Pr C	Int	Ext	Total
041UITMJ011	Imperative Programming	4	4	-	50	50	100

Course Title	IMPERATIVE PROGRAMMING
Course Credits	4
Theory Internal – External	4 Credits (Theory) 50 Marks (Internal) + 50 Marks (External)
Course Outcomes	Learners will be able to 1. Gain an understanding of the basic structure of programming languages like data types, control structures, naming conventions. 2. Understand different language paradigms like functional, imperative.
Module 1 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to- 1. Understand the basics structure of Programming Language (C language) 2. Learn to draw flowcharts and write Algorithms which helps them to develop logic.
Content Outline	1.1. Introduction: Types of Programming languages, History, features and application. Simple program logic, program development cycle, pseudocode statements and flowchart symbols, sentinel value to end a program, programming and user environments, evolution of programming models., desirable program characteristics. 1.2. Fundamentals: Structure of a program. Compilation and Execution of a Program, Character Set, identifiers and keywords, data types, constants, variables and arrays, declarations, expressions, statements, Variable definition, symbolic constants.
Module 2 (Credit 1)	

Learning Outcomes	After learning the module, learners will be able to 1. Demonstrate basic knowledge of programming using arithmetic and conditional operators in program. And also associate it with built in functions. 2. Apply and interpret the condition checking, decision making and looping by using various control structures in C programming.
Content Outline	2.1.Operators and Expressions: Arithmetic operators, unary operators, relational and logical operators, assignment operators, assignment operators, the conditional operator, library functions. 2.2.Data Input and output: Single character input and output, entering input data, scanf function, printf function, gets and puts functions, interactive programming. 2.3.Conditional Statements and Loops: Decision Making Within A Program, Conditions, Relational Operators, Logical Connectives, If Statement, If-Else Statement, Loops: While Loop, Do While, For Loop. Nested Loops, Infinite Loops, Switch Statement
Module 3 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to- 1. Analyse the concept of an array and categorize different macros by applying them in program. 2. Learn how to declare and initialize functions 3. Understand how to allocate and free memory space 4. Understand the need of file and the different file operations in C
Content Outline	1.1.Functions: Overview, defining a function, accessing a function, passing arguments to a function, specifying argument data types, function prototypes, recursion, modular programming and functions, standard library of c functions, prototype of a function: foollal parameter list, return type, function call, block structure, passing arguments to a function: call by reference, call by value. 1.2.Program structure: Storage classes, automatic variables, external variables, static variables, multifile programs, more library functions. 1.3.Preprocessor: Features, #define and #include, Directives and Macros.

	1.4. Arrays: Definition, processing, passing arrays to functions, multidimensional arrays, arrays and strings.
Module 4 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to 1. Summarize with the knowledge of pointers, file handling in c programming. 2. Understand the concepts of Structure & Union in C Programming
Content Outline	4.1.Pointers: Fundamentals, declarations, Pointers Address Operators, Pointer Type Declaration, Pointer Assignment, Pointer Initialization, Pointer Arithmetic, Functions and Pointers, Arrays and Pointers, Pointer Arrays, passing functions to other functions. 4.2.Dynamic Memory Allocation: Introduction, Allocating Block of Memory, Introduction to the following functions - Calloc(), Malloc(), Free(), Realloc. 4.3.Structures and Unions: Structure Variables, Initialization, Structure Assignment, Nested Structure, Structures and Functions, Structures and Arrays: Arrays of Structures, Structures Containing Arrays, Unions, Structures and pointers. 4.4.File Handling: Why we need a file, File operations (create, open, read, move, write, close), File opening Mode, closing a file, Input/output operations, Creating and reading a file

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Evaluation	Details	Marks
Internal	Unit test	25 marks
	Internal Assessments (Class test, Assignment)	25 marks
Total		50 marks
External	Final Exam	50 marks
Total		100 marks

REFERENCE BOOKS:

1. Farrell, J. (2014). *Programming Logic and Design*. Cengage Learning.

2. Gottfried, B. (1996). *Programming with C*. Tata McGRAWHill.
3. Kanetkar, Y. P. (n.d.). *Let us C*. BPB Publication.
4. Mothe, M. (2008). *C for Beginners*. X-Team Series.
5. Ritchie, B. W. (n.d.). *"C" Programming*. PHI.

1.2: Major (Core)

Course Code	Course	TC	Th C	Pr C	Int	Ext	Total
041UITMJ012	Operating System	2	2	-	25	25	25

Course Title	OPERATING SYSTEM
Course Credits	2
Internal – External	25 Marks + 25 Marks
Course Outcomes	After going through the course, learners will be able to • Gain knowledge about the various features of processes • Understand mechanisms of OS • Evaluate criteria for selecting a CPU-scheduling algorithm for a particular system • Study different Memory Management Strategies
Module 1 (Credit 1)	
Learning Outcome	Students will understand the Mechanisms of OS and how to handle Processes and Threads and their Communication.
Content Outline	INTRODUCTION TO OPERATING SYSTEM 1.1 Overview of operating systems, functionalities and characteristics of OS. Hardware concepts related to OS, CPU states, I/O channels 1.2 Types of OS – (Explain concepts): Single processor systems, Uni-programmed, Multiprogrammed, Batch, Timesharing-Interactive, Multitasking, Multiprocessor systems, Parallel systems, Distributed systems, Special purpose systems, Real Time systems, Multimedia systems Handheld Systems

	1.3 Process concepts: operations on processes, process states, concurrent processes, process control block, process context
Module 2 (Credit 1)	
Learning Outcome	Students will learn different Components and Management aspects of Memory, Concurrency Management, and Processor Scheduling.
Content Outline	INTRODUCTION TO SCHEDULING AND ITS TYPES 2.1 Job and processor scheduling, scheduling algorithms, process hierarchies. 2.2 Problems of Concurrent Processes, critical sections, mutual exclusion, synchronization, deadlock. 2.3 Memory management strategies: Basic concepts, Swapping – concept Contiguous memory allocation • Memory mapping & Protection • Memory Allocation • Fragmentation Non-contiguous memory allocation • Paging 2.3 Segmentation- Basic method Virtual Memory – concept

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Evaluation	Details	Marks
Internal	Unit test	25 marks
	Internal assessments (Practical test, class test)	25 marks
Total		25 marks
External	Final Exam	25 marks
Total		50 marks

REFERENCE BOOKS:

- 1) Abraham Silberschatz, Peter B. Galvin and Greg Gagne (2018). *Operating System Concepts* (10th ed.).
Dr. B.M.N. College of Home Science

- 2) Stallings, W. (2017). *Operating Systems- Internals and Design Principles* (9th ed.). Pearson.
- 3) Tanenbaum, A. S. (2016). *Modern Operating Systems* (4th ed.). Herbert Bos.
- 4) Chauhan, N. (2014). *Principles of Operating Systems*.
- 5) Godbole, & Kahate. (2010). *Operating Systems* (3rd ed.). McGraw Hill.
- 6) Goel, A. (2010). *Computer Fundamentals*.

1.3. OEC

Course Code	Course	TC	Th C	Pr C	Int	Ext	Total
041UITOE013	Digital Electronics and Logical Circuits	4	4	-	50	50	100

Course Title	DIGITAL ELECTRONICS AND LOGICAL CIRCUITS
Course Credits	4
Internal – External	25 Marks + 25 Marks
Course Outcomes	After going through the course, learners will be able to
	1. Understand the basic building blocks of digital systems. 2. To prepare students for more advanced hardware-related courses. 3. Study digital number systems and coding techniques. 4. Apply digital circuit concepts in real-world systems like digital displays, timers, and memory systems.
Module 1 (credit 1)	
Learning Outcome	Students will understand how to work with different number systems and digital codes used in computers.

Content Outline	Number system and Codes 1. Introduction: Digital Electronics vs Analog Electronics 2. Number Systems: Binary, Decimal, Octal, Hexadecimal 3. Conversion between number systems & Binary Arithmetic: Addition & Subtraction 4. 1's and 2's complement method Codes: BCD, Gray Code, ASCII & Introduction to Error Detection
Module 2 (Credit 1)	
Learning Outcome	Students will learn how logical decisions are made in digital systems, without getting into hardware gate design.
Content Outline	Basic Logic Concepts 1. Introduction to Boolean logic & Concept of logic in computers (true/false, 0/1) 2. Digital vs Analog data representation & Overview of logic blocks 3. TTL, CMOS logic families: characteristics, speed, power consumption 4. Noise margins, Fan-in, Fan-out, and Propagation delay.
Module 3	
Learning Outcome	Students will understand what memory is, types of memory, and their applications in simple terms.
Content Outline	Digital Display Systems 1. Introduction to display technologies & LED and Seven-Segment Display. 2. How numbers are shown using digital displays & Everyday examples: Calculators, Clocks, Microwave timers 3. Binary to display conversion 4. Introduction to Alphanumeric and Dot-Matrix Displays.

Module 4 (Credit 1)	
Learning Outcome	Students will understand what memory is, types of memory, and their applications in simple terms.
Content Outline	Sequential Circuits and Memory Systems 1. Flip-Flops and Latches 2. Counters 3. Shift Registers 4. Introduction to Memory Systems 5. Memory Applications

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Evaluation	Details	Marks
Internal	Unit test	25 marks
	Internal assessments (Practical test, class test)	25 marks
Total		50 marks
External	Final Exam	50 marks
Total		100 marks

REFERENCE BOOKS:

1. "Digital Fundamentals" – Thomas L. Floyd
2. "Basic Digital Electronics" – Tokheim
3. "Digital Logic and Computer Design" – M. Morris Mano (Optional)
4. "Principles of Electronics" – V.K. Mehta

1.4. VSC

Course Code	Course	TC	Th C	Pr C	Int	Ext	Total
041UITVS014	Imperative Programming LAB	2	1	1	25	25	50

Course Title	IMPERATIVE PROGRAMMING LAB
Course Credits	2
Theory Internal – External	2 Credits (Practical) 25 Marks (Internal) + 25 Marks (External)
Course Outcomes	Learners will be able to
	1. Gain an understanding of the basic structure of programming languages like data types, control structures, naming conventions.
	2. Develop the skill of writing Algorithms and drawing Flowcharts
	3. Develop the skills of C Programming implementing various concepts like functions, pointers, structures & unions
Module 1 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to- 1. Understand the basics structure of Programming Language (C language) 2. Demonstrate basic knowledge of programming using arithmetic and conditional operators in program. 3. Learn the different decision - making and loop control structures. 4. Apply and interpret the condition checking, decision making and looping by using various control structures in C programming.
Content Outline	1.1.Basic algorithms, flowchart and pseudocode: Algorithms, Flowcharts and pseudocode for implementing basic and simple programs. 1.2.Basic Programs and Programs on variables: Programs implementing different datatypes, operators, variables and built in functions. 1.3.Conditional statements and loops: Programs implementing if, nested if, if-else-if, switch, Nested switch, for loop, while loop, do-while loop. 1.4.Patterns using loops: Programs implementing different patterns. 1.5.Functions and Recursive functions: Defining and accessing, passing arguments, Function prototypes, Function calling mechanism.

Module 2 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to 1. Understand the working of functions in C and Learn to execute different programs using pre-processor directives. 2. Analyse the concept of an array and categorize different macros by applying them in program and Study how to declare and initialize a pointer and use pointers with Arrays, Functions. 3. Understand the need of file and different file operations in C 4. Understand the concepts of Structure & Union in C Programming
Content Outline	a. Concept of header files, Preprocessor directives: #include, #define and macros implementations. b. Arrays: Implementation of 1D, 2D and multidimensional arrays c. Pointers: Programs for Pointer Declaration, Initialization of Pointer Variables, Call by Value and Reference, Pointers with Arrays. d. Dynamic Memory Allocation: Programs with following functions - Calloc(), Malloc(), Free(), Realloc() e. Structures and Unions: Programs to implement Defining and Declaring Structure Variables, Dot Operator, Nested Structure, Array of Structure f. File Handling: File Operations (create, open, read, move, write, close), Program to Create a file, Program to Open a file, Program to Close a file, Input/Output operations on files

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Evaluation	Details	Marks
Internal	Internal Assessments (Practical Test, Class test)	25 marks
Total		25 Marks
External	Final Practical Exam	25 marks
Total		50 marks

REFERENCE BOOKS:

1. Farrell, J. (2014). *Programming Logic and Design*. Cengage Learning.
2. Gottfried, B. (1996). *Programming with C*. Tata McGRAWHill.
3. Kanetkar, Y. P. (2017). *Let Us C* (16th ed.). BPB Publications.
4. Mothe, M. (2008). *C for Beginners*. X-Team Series.
5. Ritchie, B. W. (n.d.). *"C" Programming*. PHI.
6. Ravi, S. (2006). *Programming Languages: Concepts and Constructs*. Pearson Education.
7. Seyed, R. (2009). *Foundations of Programming Languages Design & Implementation* (3rd ed.). Cengage learning.
8. Shah, J. (2010). *Programming in C*. Charotar Publisher.

LAB PROGRAMS:

- 1) Write a program to display the message HELLO WORLD.
- 2) Write a program to declare some variables of type int, float and double. Assign some values to these variables and display these values.
- 3) Write a program to find the addition, subtraction, multiplication and division of two numbers.
- 4) Write a program to swap two numbers without using third variable.
- 5) Write a program to find the area of rectangle, square and circle.
- 6) Write a program to find the volume of a cube, sphere, and cylinder.
- 7) Write a program to enter a number from the user and display the month name. If number >13 then display invalid input using switch case.
- 8) Write a program to check whether the number is even or odd.
- 9) Write a program to check whether the number is positive, negative or zero.
- 10) Write a program to check whether the year is a leap year or not.
- 11) Write a program to find the factorial of a number.
- 12) Write a program to check whether the entered number is prime or not.
- 13) Write a program to find the largest of three numbers.
- 14) Write a program to find the sum of squares of digits of a number.
- 15) Write a program to reverse the digits of an integer.
- 16) Write a program to find the sum of numbers from 1 to 100.
- 17) Write a program to print the Fibonacci series.
- 18) Write a program to find the reverse of a number.
- 19) Write a program to find whether a given number is palindrome or not.
- 20) Write a program that solve the quadratic equation

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- 21) Write a program to check whether the entered number is Armstrong or not.
- 22) Write a program to count the digit in a number.
- 23) Write a program in C to print the following Patterns:

*****	*	*	*****	1
****	**	**	****	2 3
***	***	***	***	4 5 6
**	****	****	**	7 8 9 0
*	*****	*****	*	

- 24) Call a function which calculates & displays maximum of 2 numbers.

- 25) Call a function which calculates & displays minimum of 3 numbers.
- 26) Call a function which calculates and displays factorial of a number.
- 27) Pass a single value from an integer array of size 5 to a function and display the same.
- 28) Pass an entire integer array containing age of 5 people to a function which calculates and displays the average age in main function.
- 29) Pass an entire 2 dimensional array to a function and display the elements of the array in the matrix form.
- 30) Write a program to find the factorial of a number using recursive function.
- 31) Write a program to find the sum of natural number using recursive function.
- 32) Write a program to find the largest value that is stored in the array.
- 33) Write a program using pointers to compute the sum of all elements stored in an array.
- 34) Write a program to arrange the 'n' numbers stored in the array in ascending and descending order.
- 35) Write a program that performs addition and subtraction of matrices.
- 36) Write a program that performs multiplication of matrices.
- 37) Write a program to demonstrate the use of pointers.
- 38) Write a program to perform addition and subtraction of two pointer variables.
- 39) Create a structure book bank containing attributes book name, author name, pages, and price. Initialize & display a structure variable.
- 40) Create a structure student containing details of roll no., std., and age. Accept & display information of 5 students.
- 41) Create 2 structures student containing details – name, rollno; grade containing details – subject, marks. Make structure grade nested to structure student. Initialize and display a structure variable.
- 42) Create a structure book containing details – name, author, callno. Initialize and display a structure variable using a pointer.
- 43) Create a structure book containing details – name, author, pages, and price. Initialize a structure variable using a pointer. Pass this structure variable to a function and display.
- 44) Create a structure and a union job with details – name, salary, workerno. Display the size of this structure and union.
- 45) Create a union employee with details – name, salary, workerno, jobprofile. Initialize and display a union variable.
- 46) Create a union student with details – name, rollno, std., age. Accept & display information of 5 students.
- 47) Create a union book containing details – name, author, callno. Initialize and display a union variable using a pointer.
- 48) Create a union book containing details – name, author, pages, and price. Initialize a union variable using a pointer. Pass this union variable to a function and display.
- 49) Executing C programs on Linux:
 - a. Write a C program to Display Hello world on Linux Operating System
 - b. Write a C program to add 2 numbers and execute it on Linux Operating System
 - c. Write a C program to find largest of 2 numbers and execute it on Linux Operating System.

1.5: SKILL ENHANCEMENT COURSE (SEC)

Course Code	Course	TC	Th C	Pr C	Int	Ext	Total
041UITSE015	GNU / Linux Programming LAB	2	-	2	25	25	50

Course Title	GNU/ LINUX PROGRAMMING LAB
Course Credits	2
Theory + Practical Internal – External	Credits (1+1) 25 Marks (Internal) + 25 Marks (External)
Course Outcomes	Upon completion of this course, students will be able to:
	1. Understand the Linux operating system architecture and environment.
	2. Apply Linux shell commands for file management and process handling.
	3. Develop practical skills using Linux terminal and command-line tools.
Module 1	
Learning Outcomes	Upon completing this Module, students will be able to:
	1. Understand and explain the Unix/Linux system architecture.
	2. Execute basic Linux commands and utilities.
	3. Navigate the Linux file system using commands.
	4. Manage and manipulate files and directories.
Content Outline	Introduction: History of Linux, Distributions, Linux kernel vs distribution, Importance of Linux in the software ecosystem: web servers, supercomputers, mobile, servers. Linux computing environment: Linux Installation methods, Hands on Installation, Unix/Linux system architecture, Basic Linux commands and general-purpose utilities. File system: System hierarchy, handling ordinary files and directories, Viewing and modifying file attributes.
Module 2	
Learning Outcomes	Upon completion of this Module, students will be able to:
	1. Use the vi editor for text editing.
	2. Redirect a command's output to a file.

	3. Understand the fundamentals of shell scripting.
	4. Write and execute basic shell scripts.
Content Outline	Vi Editor: Usage of the vi editor for editing text files, Input/output redirection, Simple filters and regular expressions (grep, sort, etc.). Basic Shell Scripting: Introduction to shell programming concepts, Features and capabilities, Syntax, Writing and executing shell scripts using variables.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Evaluation	Details	Marks
Internal	Continuous Internal Assessment (25 Marks)	
	1. Online Class Test on Module 1	10 marks
	2. Practical Test on Module 2	10 marks
	3. Lab manual assignment submission	5 marks
Total		25 Marks
External	Final Practical Exam Based on Modules 1 and 2	25 marks
Total		50 Marks

TEXT BOOK:

- Das, S. (2017). *Unix Concepts and Applications*. TMH.

REFERENCE BOOKS:

- Dumas, E., & Mevel, F. (2009). *The Linux Kernel Book*. Wiley Publications Sons.
- Kanetkar, Y. P. (2008). *Unix Shell Programming*. BPB Publications.
- Ljubuncic, I. (2011). *Linux Kernel Crash Book*.
- Maclean, M. (2016). *Just Enough Linux*. Lulu.com.
- Mathews, N. (2014). *Beginning Linux Programming*. Wrox Press.
- Blum, R. (2021). *Linux for Dummies (10th Edition)*. Wiley.
- Ward, B. (2020). *How Linux Works: What Every Superuser Should Know (3rd Ed.)*. No Starch Press.
- *Official Ubuntu Book*, 8th Edition, by Matthew Helmke & Elizabeth K. Joseph with Jose Antonio Rey and Philips Ballew, Prentice Hall (2014)

ADDITIONAL RESOURCES

- Linux kernel Home: <http://kernel.org>
- Open Source Initiative: <https://opensource.org/>
- The Linux Foundation: <http://www.linuxfoundation.org/>

LIST OF PRACTICAL SESSIONS

1. Exploring Linux architecture and terminal usage
2. Basic Linux commands – ls, pwd, cd, mkdir, rmdir, touch, echo
3. File handling – cp, mv, rm, cat, more, less, head, tail
4. File attributes – chmod, chown, stat
5. Using the vi editor – editing, saving, and navigation
6. Input/output redirection – >, >>, <, |
7. Filters and pipelines – grep, sort, uniq, wc, cut, paste
8. Introduction to shell scripting – variables, read, echo, conditional statements

1.8 VEC (Option 1)

Course Code	Course	TC	Th C	Pr C	Int	Ext	Total
041UITVE018A	Digital and Technological Solutions – Accounting Applications	2	1	1	25	25	50

Course Title	ACCOUNTING APPLICATIONS
Course Credits	2
Internal – External	25 Marks + 25 Marks
Course Outcomes	• To enable the students to gain knowledge about GNUKhata /Advanced Tally. • To improve the financial thinking skills of students. • To develop students ability to identify and evaluate accounting problems with Tally and arrive at reasoned conclusions.
Module 1 : GNU Khata /ADV. TALLY	
Learning Outcome	Students will understand: • The basic concepts and objectives of Accounting • Inventory management system.

	• Point of Sale etc. • Data Backup, Export & Import Data. • Interest calculation & Multi currency.
Content Outline	GNU Khata /ADV. TALLY 1.1 Basic of Accounting: • Basic of Accounting • Tally Fundamentals • Company creation • Ledger creation • Purchase & Sales 1.2 Inventory : • Inventory • Order Processing • Invoicing • Point of Sale • Reorder Levels 1.3 Data & Payroll: • Data Backup • Export & Import Data • Cheaque Printing • Interest Calculation • Multi-Currency • Security Control • Payroll 1.4 GENERATING & PRINTING REPORTS • Financial Statement • Books & Registers • Inventory Books & Registers • Statement of Accounts • Statement of Inventory
Module 2 : Taxation with Goods and Service Tax (GST)	
Learning Outcome	Students will understand: • The method of preparing Taxation with GST as financial statements. • The basic concept of GST, CGST & SGTS etc. • The method of Refund of Tax & GST Challan.

	• VAT TDS implications.
Content Outline	2.1 Taxation with Goods and Service Tax(GST) • Introduction of GST • GST Features • CGST • SGST • IGST • Tax Rate structure & invoicing • Refund of Tax • GST Reports & Payment • GST Challan • VAT • TDS

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Evaluation	Details	Marks
Internal	Internal assessments (Practical test, Assignments)	25 marks
Total		25 marks
External	Final Exam (Practical)	25 marks
Total		50 Marks

REFERENCE BOOKS:

- 1) Dr.H.C.Mehrotra, & Prof.V.P.Agarwal. (2019). *Goods and service Tax*. Sahitya Bhawan Publications.
- 2) Nadhani, A. K. (2018). *GST Accounting with Tally*. BPB Publication.
- 3) N. Satyapal (2001). *Tally*. Kanna Book Publication.
- 4) Nadhani, K. K. (2000). *Implementing Tally*. BPB Publication.

1.8 VEC (Option 2)

Course Code	Course	TC	Th C	Pr C	Int	Ext	Total
041UITVE018A	Digital and Technological Solutions – Office Automation	2	1	1	25	25	50

Course Title	OFFICE AUTOMATION
Course Credits	2
Internal – External	25 Marks + 25 Marks
Course Outcomes	After going through the course, learners will be able to The student will be able to use Microsoft Office applications such MS Word, Excel and PowerPoint in their day-to-day job, clerical and administration staff, support staff and their managers, business owners and entrepreneurs alike.
Module 1 : Word Processor and Presentation Software	
Learning Outcome	- Students will understand to create documents that reduce the time and cost to produce them and - They will learn the essential skills needed to create, edit, and present professional presentations, along with presentation tips.
Content Outline	1 Introduction to MS Word 1.1 Create, edit, save, and collaborate document 1.2 Formatting texts and paragraphs 1.3 Working with tables, columns, and other formatting features 1.4 Creating graphics, WordArt, charts, and text flow 1.5 Developing Microsoft Word document templates 1.6 Advanced features including mail merge and proofing tools. 2 Introduction to PowerPoint 2.1 Modify presentation themes 2.2 Adding and editing text to slides 2.3 Adding new slides to a presentation 2.4 Insert shapes and SmartArt to slides 2.5 Insert images, video and audio to slides 2.6 Insert and modify tables and charts

	2.7 Insert and edit animations and slide transitions 2.8 Creating Slide Master
Module 2 : Spread Sheet	
Learning Outcome	Students will understand how to perform mathematical and logical calculations using analytical functions.
Content Outline	3 Introduction to Excel 3.1 Formulas 3.2 Insert Charts, Filter data, sort 3.3 Pivot tables 3.4 Conditional formatting 3.5 Data Analysis

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Evaluation	Details	Marks
Internal	Internal assessments (Practical test, Assignments)	25 marks
Total		25 marks
External	Final Exam	25 marks
Total		50 marks

REFERENCE BOOKS:

1. Kumar, B. (29 January 2013). Microsoft Office 2010. INDIA: V&S Publishers.
2. Microsoft Word Step by Step (2019) - Joan Lambert
3. Michael Alexander, R. K. (4 December 2018). Excel 2019 Bible Paperback. John Wiley & Sons; Illustrated edition. "Digital Logic and Computer Design" – M. Morris Mano (Optional)
4. Lowe, D. (October 16, 2018). PowerPoint 2019 For Dummies (Powerpoint for Dummies). For Dummies; 1 edition

