SYLLABUS FOR DISCIPLINE SPECIFIC COURSE (DSC) WITH COMPUTER SCIENCE

Under Three Discipline Specific Course (FYUGP)

(To be implemented from Session 2024-25)

SEM. I & II

Year	Semester	Paper Code	Paper	Credits	Periods/Week	Exam. Marks	Continuing Evaluation Internal	Continuing Evaluation Attendance
			Proposed Syllabus for Discipline Specific Course (DSC) with Computer Science Programme					
1st Year	I	DSC1	Fundamentals of Computers	3	3	60	15	5
		DSC1T	Fundamentals of Computers (Tutorial)	1	4	-	-	-
		DSC2	To be Selected from other Discipline Core	3	3			
		DSC2T/L	To be Selected from other Discipline Core	1	1/2			
		SEC1	E1- MS Excel E2-Basic Programming in Python	2	2	40	10	5
		SEC1L	E1- MS Excel (Lab) E2-Basic Programming in Python (Lab)	1	2	20	-	-
		MIN1	Student has to choose only ONE discipline from the subjects provided by the University.	4	4			
		VAC1	Student has to choose only ONE discipline from the subjects given below: 1. Environmental Education (EE)	4	4			
	II	DSC3	Programming in C	3	3	40	10	5
		DSC3L	Programming in C (Lab)	1	1	20	-	-
		DSC4	To be Selected from other Discipline Core	3	3			
		DSC4L/T	To be Selected from other Discipline Core	1	1/2			
		SEC2	E1- Cyber Security E2-MS Power Point	2	2	60/40	10	5
		SEC2L/T	E1- Cyber Security (Tutorial) E2-MS Power Point (Lab)	1	1/2	20 (E2)	-	-
		MIN2	Student will be provided the SAME discipline from the subjects selected previously as Minor.	4	4			
		AEC1	Student has to choose only ONE discipline from the subjects given below: 1. Compulsory English	4	4			
		IDC1	Student has to choose only ONE discipline from the subjects given below: 1. Climatology 2. Chemistry in Daily Life 3. Medicinal Plants 4. Mathematics in Daily Life 5. Basics of Commerce and Management 6. Basics of Economics 7. Public Administration 8. Behavioral Science 9. Great Indian Educators 10. Social Work	3	3			
		IN1	Summer Internship The Colleges are expected to network with skill development centres,	2	-	-	-	-

			vocational training institutes for facilitating student internships. Online based internships programs are also permitted in case of Computer Science (Major) students. The students must submit a certificate of completion of the internship at the end of the semester.					

NOTE:

1. Tutorials should involve problem solving session/activity related to the subject taught.

Prerequisite(s) and/or Note(s):

(1) High school Physics.
(2) Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives:

Knowledge acquired:

(1) Introduce students to the fundamental concepts of system tools, their functionalities, and their role in computer systems.
(2) Familiarize students with various peripheral devices commonly used in computer systems and their functionalities.
(3) Develop students' proficiency in utilizing system tools to optimize computer performance, troubleshoot issues, and maintain system integrity.
(4) Enable students to effectively manage peripheral devices, including printers, scanners, external storage devices, and input/output devices.

Skills gained:

(1) Basic idea programming.
(2) Troubleshooting hardware and software issues.
(3) Efficient internet and file management.

Competency Developed:

(1) Problem-solving in technical contexts.
(2) Adaptability to various software environments.
Effective communication of technical concepts.

Syllabus Overview

Unit 1:	Basics of Computer	15 Lectures

Generation of Computers; Computer system : Basic Block Diagram, Super Mainframe, Mini & Personal Computer, Nomenclature, Software : Systems and Application; Hardware & Software; Algorithms : Definition, essential features;

Unit 2:	Peripheral Devices	15 Lectures

Input and Output Devices – Punched Card, Keyboard, Mouse, Joystick, Trackball, Light Pen, Touch Screen, Magnetic Ink Character Recognition (MICR), Optical Character Recognition (OCR), Optical Mark Recognition (OMR), Display units, Printers- Impact and Non-Impact. Primary storage – RAM-SRAM, DRAM, ROM-PROM, EPROM, EEPROM, Secondary storage – Hard drive, Magnetic drive, Compact Disk, Cache memory, components of motherboard.

Unit 3:	Programming Fundamentals	10 Lectures

Complexity : notation, time & space; Computability & correctness concepts; Structured programming concepts; Process of problem solving, Flowcharts and Pseudo codes.

Unit 4:	Software and Data	5 Lectures

Concept of software, Firmware, Types of Software: (System software, Application software, Utility software), Concept of Operating System, Loaders, Linkers, Debuggers, Translators: (Compilers, Assemblers, Interpreters), Data and Information, Types of data, Units of data measurement, Qualities of Information, Concept of database, Languages, Generation of Languages.

Suggested Readings

1. Sinha P.K., "Computer Fundamentals", 6 th Edition, BPB Publication, 2012.
2. Rajaraman,V.,"Computer Fundamentals", 6 th Edition, PHI,2012.
3. Thareja R., "Fundamentals of Computers", Oxford University Press, 2014.
4. Stallings W., "Operating systems", 8th Edition, Pearson, 2014.

Course-DSC Paper:	Paper Code-DSC1T Computer Fundamentals (Tutorial)	Credits-1	Tut./Week-1

Computer Fundamentals Tutorial as assigned and advised by teacher(s).

Course- SEC Paper:	Paper Code-SEC1 E1 MS Excel	Credits-3	Lectures/Week-3

Prerequisite(s) and/or Note(s):

(1) High school mathematics.
(2) Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives:

Knowledge Acquired:

(1) Formulas and Functions: Students learn various Excel formulas and functions such as SUM, AVERAGE, and IF statements, enabling them to perform complex calculations efficiently.
(2) Data Management Techniques: They gain knowledge of sorting, filtering, and organizing data effectively within Excel spreadsheets, including techniques like data validation and conditional formatting.
(3) Data Analysis Tools: Students acquire an understanding of Excel's data analysis tools like pivot tables, charts, and what-if analysis, empowering them to derive insights and make data-driven decisions.

Skills Gained:

(1) Data Manipulation: Students develop skills in manipulating data, including tasks like merging cells, splitting data, and removing duplicates, enhancing their ability to clean and format datasets.
(2) Charting and Visualization: They learn to create visually appealing charts and graphs, mastering skills to represent data in a meaningful and comprehensible manner.
(3) Automation and Macros: Students acquire skills in automating repetitive tasks through macros, increasing productivity and efficiency in handling large datasets.

Competency Developed:

(1) Problem-Solving: Through solving various real-world data management and analysis challenges, students enhance their problem-solving abilities within Excel, learning to devise efficient solutions.
(2) Attention to Detail: Working extensively with data requires meticulous attention to detail to avoid errors. Students develop this competency through tasks like data validation and auditing.
(3) Collaboration and Communication: Excel often serves as a collaborative tool in professional settings. Students learn to collaborate effectively on spreadsheets and

communicate their findings clearly through the use of Excel's sharing and commenting features.

Syllabus Overview

Unit 1:	Manage Workbook Options and Settings	10 Lectures

Create a workbook, Import data from a delimited text file, Add a worksheet to an existing workbook, Copy and move a worksheet, Search for data within a workbook, Navigate to a named cell, range, or workbook element, Insert and remove hyperlinks, Change worksheet tab color, Rename a worksheet, Change worksheet order, Insert and delete columns or rows, Change workbook themes, Adjust row height and column width, Insert headers and footers, Hide or unhide worksheets, Hide or unhide columns and rows, Customize the Quick Access toolbar, Modify document properties, Display formulas.

Unit 2:	Creating and Managing Tables	10 Lectures

Create an Excel table from a cell range, convert a table to a cell range, Add or remove table rows and columns, apply styles to tables, configure table style options, Insert total rows, filter records sort data by multiple columns, change sort order, remove duplicate records

Unit 3:	Perform Operations with Formulas and Functions	5 Lectures

Perform calculations by using the SUM function, perform calculations by using MIN and MAX functions, perform calculations by using the COUNT function, perform calculations by using the AVERAGE function, perform Conditional Operations by using functions (IF, SUMIF, AVERAGEIF, COUNTIF)

Unit 4:	Create Charts and Objects	5 Lectures

Create a new chart (Bar, Line, Scatter plot, Pie, Area), add additional data series, switch between rows and columns in source data, resize charts, add and modify chart elements, apply chart layouts and styles, move charts to a chart sheet.

Suggested Readings

1. "Excel 2019 Bible Paperback", Michael Alexander (Author), Richard Kusleika (Author), John Walkenbach (Author)
2. "Excel for Beginners (Excel Essentials Book 1)", M.L. Humphrey (Author)
3. "Ctrl+Shift+Enter Mastering Excel Array Formulas: Do the Impossible with Excel Formulas Thanks to Array Formula Magic", Mike Girvin (Author)

Course-SEC Paper:	Paper Code-SEC1L E1 MS Excel (Lab)	Credits-1	Lab hours/Week-2

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Create a workbook and enter the raw data applying as many presentation Features (Font, Font Size, Font Colour, Number Formats and Colour, Cell Shading, Text Rotation, etc)
2. Apply appropriate number formats to your numbers.
3. Select the best page orientation for your spreadsheet.
4. Adjust the column width and row height to suit the layout you have selected.
5. Create formula's to calculate the percentage of the total number of people
6. Setup an appropriate title, header, footer and page number in your spreadsheet.
7. Create the following Table in Excel with given details :-

RollNo, Name, Math, English, Science, Total Result, Division

Course- SEC Paper:	Paper Code-SEC1 E2 Basic Programming in Python	Credits-3	Lectures/Week-3

Prerequisite(s) and/or Note(s):
(1) High school mathematics.
(2) Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives:

Knowledge Acquired:
   (1) Fundamental Concepts: Students acquire knowledge of fundamental programming concepts such as variables, data types, loops, conditionals, and functions in Python.
   (2) Data Structures: They learn about essential data structures like lists, tuples, dictionaries, and sets, understanding their usage and implementation.

Skills Gained:
   (1) Coding Proficiency: Through hands-on practice and assignments, students develop coding proficiency in Python, enabling them to write clear, concise, and functional code.
   (2) Problem-Solving: They enhance their problem-solving skills by applying Python programming concepts to solve various computational problems and algorithms.
   (3) Debugging and Troubleshooting: Students acquire skills in debugging code and troubleshooting errors, learning how to identify and fix common programming mistakes effectively.

Competency Developed:
   (1) Logical Thinking: Python programming exercises require logical thinking and algorithmic problem-solving skills, helping students develop a logical mindset.
   (2) Attention to Detail: Writing code necessitates attention to detail to ensure accuracy and functionality. Students develop this competency through debugging and code review processes.
   (3) Collaboration and Documentation: Students learn to collaborate on coding projects using version control systems like Git and to document their code effectively, enhancing their ability to work in teams and communicate technical concepts clearly.

Syllabus Overview

Unit 1:	Introduction to Python	10 Lectures

Structure of a Python Program, Elements of Python, Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables

Unit 2:	Flow control and Functions	10 Lectures

Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys.exit(),def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling.

Unit 3:	List, Dictionary, String and Tuples	10 Lectures

String, String functions, Manipulating Strings, Lists: Creating Lists; Operations on Lists; Built-in Functions on Lists; Implementation of Stacks and Queues using Lists; Nested Lists.

Dictionaries: Creating Dictionaries; Operations on Dictionaries; Built-in Functions on Dictionaries; Dictionary Methods; Populating and Traversing Dictionaries.

Tuples and Sets: Creating Tuples; Operations on Tuples; Built-in Functions on Tuples; Tuple Methods; Creating Sets; Operations on Sets; Built-in Functions on Sets; Set Methods.

Suggested Reading

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation www.python.or 2015
3. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online. 2012
4. http://docs.python.org/3/tutorial/index.html
5. http://interactivepython.org/courselib/static/pythonds
6. http://www.ibiblio.org/g2swap/byteofpython/read/

Course-SEC Paper:	Paper Code-SEC1L E2 Basic Programming in Python (Lab)	Credits-1	Lab hours/Week-2

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users' choice.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria:
    a. Grade A: Percentage >=80
    b. Grade B: Percentage>=70 and <80
    c. Grade C: Percentage>=60 and <70
    d. Grade D: Percentage>=40 and <60
    e. Grade E: Percentage<40

3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to implement the use of arrays in Python.
7. WAP to implement String Manipulation in python in Python.
8. WAP to find sum of the following series for n terms: $1 – 2/2! + 3/3! - - - - - n/n!$

Course-DSC Paper:	Paper Code-DSC3 Programming Fundamentals Using C	Credits-3	Lectures/Week-3

Prerequisite(s) and/or Note(s):
(1) High school mathematics.
(2) Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives:

Knowledge acquired:
(1) Knowledge about program development and implementation
(2) Syntax of C programming language
(3) Knowledge about how humans interact with computers through a language.

Skills gained:
(1) Problem solving skills
(2) Logical thinking to approach a problem
(3) Building programs for different problems at hand.

Competency Developed:
(1) Applying the skills learnt to model real world problems
(2) Facility in solving real life problems by thinking logically and outside of box.
(3) Ease of switching to any other programming language

Syllabus Overview

Unit 1:	Introduction to C programming	6 Lectures

History of C, Overview of Procedural Programming, Introduction to Algorithm & Flowcharts. Using main() function Compiling and Executing Simple Programs in C. Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putchar etc), Formatted and Console I/O (printf(), scanf()) , Using Basic Header Files (stdio.h, iostream.h, conio.h etc).

Unit 2:	Expression and Control Flow	6 Lectures

Simple Expressions in C (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)

Unit 3:	Functions	5 Lectures

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments.

Unit 4:	Arrays and Strings in C	5 Lectures

Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays.

Unit 5:	User-defined Data Types and Pointers Basics	6 Lectures

Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, Manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members. Pointers, integer, float and string pointers, working of pointers.

Suggested Readings
1.   Rajaraman V. & Radhakrishnan, An Introduction To Digital Computer Design, PHI.
2.   Malvino & Leach, Digital Principles & Applications, TMH
3.   S. Salivahanan, S. Arivazhagan, Digital Circuits and Design, Oxford University Press

Course-MAJOR Paper:	Paper Code-DSC3L Programming Fundamentals Using C (Lab)	Credits-1	Lab hours/Week-2

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. WAP to perform input/output of all basic data types.
2. WAP to enter two numbers and find their sum.
3. WAP to reverse a number.
4. WAP to Swap Two Numbers (using and without using a third variable).
5. WAP to check whether a number is even or odd
6. WAP to compute the factors of a given number.
7. WAP to enter marks of five subjects and calculate total, average and percentage.
8. WAP to print the sum and product of digits of an integer.
9. WAP to check whether a character is vowel or consonant
10. WAP to find the largest among three numbers

Course- SEC Paper:	Paper Code-SEC2 E1 Cyber Security	Credits-2	Lectures/Week-2

Prerequisite(s) and/or Note(s):
(1) High school mathematics.
(2) Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives:

Knowledge Acquired:
(1) Cyber threats landscape understanding.
(2) Principles of cryptography comprehension.
(3) Network security protocols familiarity.
Skills Gained:

(1) Ethical hacking techniques application.
(2) Security assessment tools utilization.
(3) Incident response plan development.

Competency Developed:
(1) Risk assessment proficiency.
(2) Security policy formulation expertise.
(3) Communication of security concepts clarity.

Syllabus Overview

Unit 1:	Introduction	5 Lectures

Introduction, Computer Security, Threats, Harm, Vulnerabilities, Authentication Mechanisms - Passwords, Biometrics, Hardware Tokens, Authorization and Access Control Lists (ACLs).

Unit 2:	Firewalls	5 Lectures

Firewalls and Packet Filters: Firewall Basics, Packet Filter Vs Firewall, How a Firewall Protects a Network, Packet Characteristic to Filter, Stateless Vs Stateful Firewalls, Network Address Translation (NAT) and Port Forwarding.

Unit 3:	Introduction to Cyber Crime, law and Investigation	10 Lectures

Cyber Crimes, Types of Cybercrime, Hacking, Attack vectors, Cyberspace and Criminal Behavior, Clarification of Terms, Traditional Problems Associated with Computer Crime, Introduction to Incident Response, Digital Forensics, Computer Language, Network Language, Realms of the Cyber world. Internet crime and Act: A Brief History of the Internet, Recognizing and Defining Computer Crime, Contemporary Crimes, Computers as Targets, Contaminants and Destruction of Data, Indian IT ACT 2000.

Unit 4:	Intrusion Detection Systems (IDS)	10 Lectures

Overview and Importance, Types of IDS: Host-Based IDS, Network-Based IDS, IDS Architectures: Centralized IDS, Distributed IDS, Detection Techniques: Signature Based, Statistical, Anomaly Detection Based (Various Features like User Login Time, Duration etc.), IDS Configuration and Management: IDS Sensor, Configuration and Rule Creation using Snort

Suggested Readings
4. "Cybersecurity for Dummies" by Chey Cobb.
5. "Computer Hacking Beginners Guide" by Alan T. Norman
6. "Hacking: Computer Hacking, Security Testing, Penetration Testing, and Basic Security" by John Slavio

Course-SEC Paper:	Paper Code-SEC1T E1 Cyber Security (Tutorial)	Credits-1	Tut. hours/Week-1

Cyber Security Tutorial as assigned and advised by teacher(s).

Course- SEC Paper:	Paper Code-SEC2 E2 MS Power Point	Credits-2	Lectures/Week-2

Prerequisite(s) and/or Note(s):
(1) High school mathematics.
(2) Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives:

Knowledge Acquired:
(1) Presentation design principles understanding.
(2) MS PowerPoint interface familiarity.
(3) Slide layout and formatting comprehension.

Skills Gained:
(1) Slide creation and editing proficiency.
(2) Visual content insertion capability.
(3) Animation and transition application skill.

Competency Developed:
(1) Effective presentation delivery competency.
(2) Audience engagement techniques mastery.
(3) Time management during presentations efficiency.

Syllabus Overview

Unit 1:	Creating and Managing Presentations	10 Lectures

Create a Presentation: Insert and Format Slides, Modify Slides, Handouts, and Notes, Change Presentation Options and Views, Configure a Presentation for Print, Configure and Present a Slide Show, Insert and Format Text: Insert and Format Shapes and Text Boxes, Insert and Format Images, Order and Group Objects.

Unit 2:	Tables, Charts, SmartArt, and Media	10 Lectures

Insert and Format Tables: Insert and Format Charts, Insert and Format SmartArt graphics, Insert and Manage Media.

Unit 3:	Transitions and Animations	10 Lectures

Apply Slide Transitions, Animate Slide Content, Set Timing for Transitions and Animations, Working with bullets and numbering, Working with different views, Working with slide Master, Slide show option

Suggested Reading
1. Microsoft power point 2019 ,learning the basics by Eric Stockson
2. Microsoft power point 2019 for beginners by J.Davidson.
3. Marquee series MIcrosoft power point 2019 by Audrey Roggenkamp & Lan Rutkowsky ,Nita Rutkosky

Course-SEC Paper:	Paper Code-SEC2L E2 MS Power Point (Lab)	Credits-1	Lab hours/Week-2

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

(1) Creating a Title Slide
(2) Creating Slides Using Layouts
(3) Create a presentation that consists of 5 slides and save your Presentation in desktop.
(4) Demonstrate slide transitions and animation
(5) Insert slide number, slide date, slide header and footer
(6) Demonstrate rehearse time.
(7) Demonstrate master slide.