SYLLABUS FOR COMPUTER SCIENCE (MAJOR)

Under Single Major Single Minor (FYUGP)

(To be implemented from Session 2024-25)

SEM. I & II

Proposed Syllabus for four years B.Sc. Computer Science (Major) Programme								
Year	**Semester**	**Paper Code**	**Paper**	**Credits**	**Periods/Week**	**Exam. Marks**	**Continuing Evaluation**	
							Internal	**Attendance**
1st Year	**I**	MAJ1	Digital Design and Analysis	3	3	40	10	5
		MAJ1L	Digital Design and Analysis (Lab)	1	4	20	-	-
		MAJ2	Programming in C	3	3	40	10	5
		MAJ2L	Programming in C (Lab)	1	4	20	-	-
		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 given below: 1. Physics 2. Mathematics 3. Statistics 4. Economics 5. Geography	4	4	40 /60	10	5
		VAC1	Student has to choose only ONE discipline from the subjects given below: 1. Environmental Education (EE)	4	4	60	10	5
	II	MAJ3	Discrete Structures	3	3	60	10	5
		MAJ3T	Discrete Structures (Tutorial)	1	1	-	-	-
		MAJ4	Object Oriented Programming Using Java	3	3	40	10	5
		MAJ4L	Object Oriented Programming Using Java (Lab)	1	2	20	-	-
		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	40 /60	10	5
		AEC1	Student has to choose only ONE discipline from the subjects given below: 1. Compulsory English	4	4	30	15	5
		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	3	3	60	10	5

			9. Great Indian Educators 10. Social Work					
		IN1	Summer Internship The Colleges are expected to network with skill development centres, 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.	2	-	-	-	-

NOTE:

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

1st Year
Semester-I

Course-MAJOR	**Paper Code-MAJ1**	**Credits-3**	**Lectures/Week-3**
Paper:	**Digital Design and Analysis**		

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) Basic knowledge of digital logic and digital circuits,
(2) Overall idea about how computers function and the internal building blocks of a computer.
(3) Knowledge about how operations are performed in a computer
(4) A thorough understanding of the fundamental concepts and techniques used in digital electronics.

Skills gained:
(1) Application of the knowledge of digital logic to understand digital electronics circuits.
(2) The ability to understand, analyze and design various combinational and sequential circuits.
(3) To understand and examine the structure of various number systems and its application in digital design.

Competency Developed:
(1) Ability to identify basic requirements for a design application and propose a cost effective solution.
(2) The ability to identify and prevent various hazards and timing problems in a digital design.
(3) Ability and skill to develop/build, and troubleshoot digital circuits.

Syllabus Overview

Unit 1: Fundamentals of Computers **7 Lectures**

Generation of Computers and Computer Languages, Computer Systems, Basic block Diagram, Von-Neumann Architecture, Types of Computers, Hardware, Firmware, I/O Devices, Storage classifications, Language translators.

Unit 2: Number Systems and Codes **10 Lectures**

Binary, octal, hexadecimal and decimal number systems and their inter conversion, BCD numbers (8421-2421), Gray code, excess–3 code, code conversion, ASCII, EBCDIC codes, their advantages and disadvantage, Binary addition and subtraction, Negative number representation: Sign magnitude, 1's, 2's Complement. signed and unsigned binary numbers, Fixed and floating-point representation.

Unit 3: Logic Gates **7 Lectures**

AND, OR, NOT Gates and their Truth Tables, NOR, NAND & XOR gates, Boolean algebra, Basic Boolean Laws, De-morgan's theorem, Boolean function and their truth tables, Minimization techniques, K-Map for 2, 3 and 4 variables, Sum of Product & Product of Sum, Don't care conditions.

Unit 4: Logic Families **7 Lectures**

Introduction to digital logic family such as RTL, DTL, TTL, ECL, CMOS, IIR, HTL etc., their comparative study, Basic circuit, performance characteristics.

Unit 5: Combinational Logic **7 Lectures**

Half adder, Full adder, parallel adder, half subtractor, full subtractor, 4-bit binary adder cum subtractor, Multiplexer, Demultiplexer, Decoder, BCD to seven segment Decoder, Encoders.

Unit 6: Sequential Circuit: **7 Lectures**

Set-reset latches, D-flip-flop, R-S flip-flop, J-K flip-flop, Master slave flip-flop, edge triggered flip-flop, T flip-flop, Synchronous/Asynchronous counter, Up/down synchronous counter, Ripple Counter, Applications of counter, Serial in/Serial out shift register, Parallel in/Serial out shift register, Serial in/parallel out shift register, parallel in/ parallel out shift register, Bi-directional register, Applications of register.

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 Code-MAJ1L Credits-1 Lab hours/Week-2
Paper: Digital Design and Analysis (Lab)

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

1. Design of a XOR gate using basic gates.
2. Design of an 8x1 MUX using basic gates.
3. Design of a Half Adder using basic gates.
4. Design of a Full Subtractor using basic gates.
5. Design of a 2-to-4 Decoder using basic gates.

Course-MAJOR Paper:	Paper Code-MAJ2 Programming in 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, Data Types, Variables and Operators 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: Expressions, Conditional Statements and Iterative Statements 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: Understanding 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: Implementation of Arrays and Strings 6 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 (Structures and Unions) 5 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.

Unit 6: Pointers and References in C 10 Lectures

Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with Pointers, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values. Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, use of new and delete operators, storage of variables in static and dynamic memory allocation.

Unit 7: File and I/O 7 Lectures

Opening and closing a file, Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files.

Suggested Readings

1. "The C Programming Language ANSI C Version", Kernighan & Ritchie, Prentice Hall Software Series
2. "ANSI C - Made Easy", Herbert Schildt, Osborne McGraw-Hill
3. "Learning to Program in C", N. Kantaris, Babani
4. "C - The Complete Reference", Herbert Schildt, Osborne McGraw-Hill
5. "Programming in C", Reema Thareja, Oxford University Press
6. "A First Course in Programming With C", T. Jeyapoovan, Vikas Publishing House
7. "Let Us C", Yashavant P. Kanetkar, BPB Publications

Course-MAJOR	**Paper Code-MAJ2L**	**Credits-1**	**Lab hours/Week-2**
Paper:	**Programming in C (Lab)**		

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

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.

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

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

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 Readings

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

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!

1st Year Semester-II			
Course-MAJOR Paper:	**Paper Code-MAJ3 Discrete Structures**	**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) Basic knowledge of discrete mathematics and discrete structures,
(2) To develop understanding of Logic sets and functions
(3) Knowledge of mathematically correct terminology and notations.
(4) Knowledge about construction of direct and indirect proofs.

Skills gained:
(1) Development of problem-solving skills necessary for understanding counting problems.
(2) Ability to generalize from a single instance of a problem an entire class of problems and identification of patterns of data.

Competency Developed:
. (1) Ability to analyze problems and solve problems.
(2) Ability to implement mathematical knowledge in data analysis..

Syllabus Overview

Unit 1: Introduction **10 Lectures**

Sets - finite and Infinite sets, uncountably Infinite Sets; functions, relations, Properties of Binary Relations, Closure, Partial Ordering Relations; counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion.

Unit 2: Growth of Functions **8 Lectures**

Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals.

Unit 3: Recurrence Relations **10 Lectures**

Basic Terminology, Models and Types, multigraphs and weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and properties of Trees, Introduction to Spanning Trees, graph traversals (BFS, DFS).

Unit 4: Graph Theory **10 Lectures**

Basic Terminology, Models and Types, multigraphs and weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and properties of Trees, Introduction to Spanning Trees, graph traversals (BFS, DFS).

Unit 5: Propositional Logic **7 Lectures**

Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory, Quantifiers.

Suggested Readings

1.C.L. Liu , D.P. Mahopatra, Elements of Discrete mathematics, 2nd Edition , Tata McGraw Hill, 1985,

2.Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition ,McGraw Hill 2006

3.T.H. Coremen, C.E. Leiserson, R. L. Rivest, Introduction to algorithms, 3rd edition Prentice Hall on India, 2009

4.M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms , John wiley Publication, 1988

5.J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Edition, Jones and Bartlett Publishers, 2009

6.D.J. Hunter, Essentials of Discrete Mathematics, Jones and Bartlett Publishers, 2008

7.Discrete Mathematical Structures with Applications to Combinatorics, V Ramaswamy, University Press

8.Discrete Mathematics: A Concept-based Approach, Basavaraj S Anami, Venkanna S Madalli, University Press

Course-MAJOR Paper:	**Paper Code-MAJ3T Discrete Structures (Tutorial)**	**Credits-1**	**Tut./Week-1**

Discrete Structures Tutorial as assigned and advised by teacher(s).

Course-MAJOR	Paper Code-MAJ4	Credits-3	Lectures/Week-3
Paper:	Object Oriented Programming Using Java		

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) Understanding of Object-Oriented Concepts: Students will acquire knowledge of fundamental Object-Oriented Programming (OOP) concepts such as classes, objects, inheritance, polymorphism, and encapsulation. They'll grasp the theoretical underpinnings of these concepts and their practical applications in software development.

(2) Java Syntax and Language Features: Through hands-on coding exercises and projects, students will become proficient in Java syntax, learning about data types, control flow structures, and exception handling. They'll understand how to write Java programs that follow best practices and adhere to industry standards.

(3) Software Design Principles: The course will equip students with knowledge of software design principles like SOLID principles, design patterns, and anti-patterns. They'll learn how to architect well-structured, maintainable, and extensible software systems using object-oriented design principles.

Skills gained:

(1) Programming Proficiency: Students will develop practical programming skills in Java, including the ability to write, compile, and execute Java programs independently. They'll gain confidence in coding by solving progressively challenging programming problems and implementing real-world applications.

(2) Debugging and Troubleshooting: Through debugging exercises and code reviews, students will learn how to identify and fix errors in Java code effectively. They'll develop skills in using debugging tools and techniques to diagnose and resolve software issues efficiently.

Competency Developed:

(1) Problem-Solving Skills: Students will enhance their problem-solving abilities by applying object-oriented principles to solve complex programming problems. They'll learn how to break down problems into smaller, manageable components and devise elegant solutions using OOP concepts.

(2) Critical Thinking and Analysis: The course will foster students' ability to critically evaluate software designs and code implementations. They'll learn to analyze trade-offs, identify design flaws, and propose alternative solutions, honing their critical thinking skills essential for software development.

(3) Software Development Practices: By working on practical projects, students will develop competency in software development practices such as unit testing, code documentation, and code refactoring. They'll understand the importance of writing clean, readable, and maintainable code, preparing them for careers in software engineering or further academic pursuits in computer science.

Syllabus Overview

Unit 1: Introduction to Java **4 Lectures**

Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods).

Unit 2: Arrays, Strings and I/O **7 Lectures**

Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

Unit 3: Object -Oriented Programming Overview **7 Lectures**

Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

Unit 4: Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata **10 Lectures**

Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

Unit 5: Exception Handling, Threading, Networking and Database Connectivity **6 Lectures**

Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

Unit 6: Applets and Event Handling **10 Lectures**

Java Applets:Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms,Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, textfields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.

Suggested Readings

1.Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.
2.James Gosling, Bill Joy, Guy L Steele Jr, GiladBracha, Alex Buckley"The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.
3.Joshua Bloch, "Effective Java" 2nd Edition,Publisher: Addison-Wesley, 2008.
4.Cay S. Horstmann, GaryCornell, "Core Java 2 Volume 1 ,9th Edition,Printice Hall.2012
5.Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 2 - Advanced Features)", 9th Edition, Printice Hall.2013
6.Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
7.E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.
8.Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall, 2011.

Course-MAJOR	**Paper Code-MAJ2L**	**Credits-1**	**Lab hours/Week-2**
Paper:	**Object Oriented Programming Using Java (Lab)**		

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

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of ".length" in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument

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

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.
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 **5 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) **5 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

1. "Cybersecurity for Dummies" by Chey Cobb.
2. “Computer Hacking Beginners Guide” by Alan T. Norman

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 5 Lectures

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

Unit 3: Transitions and Animations 5 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 Readings

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.