



---

# SYLLABUS FOR BCA (MAJOR)

---

Under Single Major Single Minor (FYUGP)  
(To be implemented from Session 2024-25)

SEM. I & II

Proposed Syllabus for four years B.C.A. (Major) Programme								
Year	Semester	Paper Code	Paper	Credits	Periods/Week	Exam. Marks	Continuing Evaluation	
							Internal	Attendance
1 <sup>st</sup> Year	I	MAJ1	Computer Fundamentals	3	3	60	10	5
		MAJ1L	Computer Fundamentals (Tutorial)	1	4	-	-	-
		MAJ2	Programming Fundamentals Using C	3	3	40	10	5
		MAJ2L	Programming Fundamentals Using 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
		MAJ3	Digital Electronics	3	3	-	-	-
		MAJ3T	Digital Electronics (Lab)	1	1	40	10	5
	MAJ4	Java Programming	3	3	20	-	-	
	MAJ4L	Java Programming (Lab)	1	2	60 /40	10	5	
	SEC2	E1- Cyber Security E2-MS Power Point	2	2	20 (E2)	-	-	
	SEC2L/T	E1- Cyber Security (Tutorial) E2-MS Power Point (Lab)	1	1/2	40 /60	10	5	
	MIN2	Student will be provided the SAME discipline from the subjects selected previously as Minor.	4	4	30	15	5	
	AEC1	Student has to choose only ONE discipline from the subjects given below: 1. Compulsory English	4	4	60	10	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	3	3				

			8. Behavioral Science 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.

1 <sup>st</sup> Year Semester-I			
Course-MAJOR Paper:	Paper Code-MAJ1 Computer Fundamentals	Credits-3	Lectures/Week-3

**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.
- (3) Effective communication of technical concepts.

**Syllabus Overview**

<b>Unit 1:</b>	<b>Basics of Computer</b>	<b>15 Lectures</b>
Generation of Computers; Computer system : Basic Block Diagram, Super Mainframe, Mini & Personal Computer, Nomenclature, Software : Systems and Application; Hardware & Software; Algorithms : Definition, essential features;		
<b>Unit 2:</b>	<b>Peripheral Devices</b>	<b>15 Lectures</b>
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.		
<b>Unit 3:</b>	<b>Programming Fundamentals</b>	<b>10 Lectures</b>
Complexity : notation, time & space; Computability & correctness concepts; Structured programming concepts; Process of problem solving, Flowcharts and Pseudo codes.		
<b>Unit 4:</b>	<b>Basics of Operating Systems</b>	<b>5 Lectures</b>
Introduction to Operating Systems (Disk Operating System, Windows, Unix, Linux), System Administration.		

### 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.

<b>Course-MAJOR</b>	<b>Paper Code-MAJIT</b>	<b>Credits-1</b>	<b>Tut./Week-1</b>
---------------------	-------------------------	------------------	--------------------

<b>Paper:</b>	<b>Computer Fundamentals (Tutorial)</b>
---------------	---

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

<b>Course-MAJOR</b>	<b>Paper Code-MAJ2</b>	<b>Credits-3</b>	<b>Lectures/Week-3</b>
---------------------	------------------------	------------------	------------------------

<b>Paper:</b>	<b>Programming Fundamentals Using C</b>
---------------	---

### *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

<b>Unit 1:</b>	<b>Introduction to C programming</b>	<b>6 Lectures</b>
----------------	--------------------------------------	-------------------

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).

<b>Unit 2:</b>	<b>Expression and Control Flow</b>	<b>6 Lectures</b>
----------------	------------------------------------	-------------------

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) .

<b>Unit 3:</b>	<b>Functions</b>	<b>5 Lectures</b>
----------------	------------------	-------------------

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.

<b>Unit 4:</b>	<b>Arrays and Strings in C</b>	<b>6 Lectures</b>
----------------	--------------------------------	-------------------

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.

<b>Unit 5:</b>	<b>User-defined Data Types</b>	<b>5 Lectures</b>
----------------	--------------------------------	-------------------

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.

<b>Unit 6:</b>	<b>Pointers in C</b>	<b>10 Lectures</b>
----------------	----------------------	--------------------

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.

<b>Unit 7:</b>	<b>I/O and File Systems in C</b>	<b>7 Lectures</b>
----------------	----------------------------------	-------------------

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

<b>Course-MAJOR</b>	<b>Paper Code-MAJ2L</b>	<b>Credits-1</b>	<b>Lab hours/Week-2</b>
<b>Paper:</b>	<b>Programming Fundamentals Using C (Lab)</b>		

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

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

***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

<b>Unit 1:</b>	<b>Manage Workbook Options and Settings</b>	<b>10 Lectures</b>
----------------	---	--------------------

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.

<b>Unit 2:</b>	<b>Creating and Managing Tables</b>	<b>10 Lectures</b>
----------------	-------------------------------------	--------------------

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

<b>Unit 3:</b>	<b>Perform Operations with Formulas and Functions</b>	<b>5 Lectures</b>
----------------	---	-------------------

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)

<b>Unit 4:</b>	<b>Create Charts and Objects</b>	<b>5 Lectures</b>
----------------	----------------------------------	-------------------

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)

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

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*



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

**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**

<b>Unit 1:</b>	<b>Introduction to Python</b>	<b>10 Lectures</b>
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		

<b>Unit 2:</b>	<b>Flow control and Functions</b>	<b>10 Lectures</b>
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.		

<b>Unit 3:</b>	<b>List, Dictionary, String and Tuples</b>	<b>10 Lectures</b>
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.org](http://www.python.org) 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
<b>Paper: Basic Programming in Python (Lab)</b>			

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  $\geq 80$
  - b. Grade B: Percentage  $\geq 70$  and  $< 80$
  - c. Grade C: Percentage  $\geq 60$  and  $< 70$
  - d. Grade D: Percentage  $\geq 40$  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:  $1 - 2/2! + 3/3! - \dots - n/n!$

**1<sup>st</sup> Year  
Semester-II**

<b>Course- MAJOR Paper:</b>	<b>Paper Code-MAJ3 Digital Electronics</b>	<b>Credits-3</b>	<b>Lectures/Week-3</b>
---------------------------------	--	------------------	------------------------

***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**

<b>Unit 1:</b>	<b>Number Systems</b>	<b>10 Lectures</b>
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.		
<b>Unit 2:</b>	<b>Gates in Digital Design</b>	<b>10 Lectures</b>
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.		
<b>Unit 3:</b>	<b>Logic Families</b>	<b>7 Lectures</b>
Introduction to digital logic family such as RTL, DTL, TTL, ECL, CMOS, IIR, HTL etc., their comparative study, Basic circuit, performance characteristics.		

<b>Unit 4:</b>	<b>Combinational Logic</b>	<b>8 Lectures</b>
----------------	----------------------------	-------------------

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.

<b>Unit 5:</b>	<b>Sequential Logic</b>	<b>10 Lectures</b>
----------------	-------------------------	--------------------

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

<b>Course-MAJOR</b>	<b>Paper Code-MAJ3L</b>	<b>Credits-1</b>	<b>Lab hours/Week-2</b>
<b>Paper:</b>	<b>Digital Electronics (Lab)</b>		

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 a 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.

<b>Course-MAJOR</b>	<b>Paper Code-MAJ4</b>	<b>Credits-3</b>	<b>Lectures/Week-3</b>
<b>Paper:</b>	<b>Java Programming</b>		

**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

<b>Unit 1:</b>	<b>Introduction to Java</b>	<b>4 Lectures</b>
----------------	-----------------------------	-------------------

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).

<b>Unit 2:</b>	<b>Arrays, Strings and I/O</b>	<b>7 Lectures</b>
----------------	--------------------------------	-------------------

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.

<b>Unit 3:</b>	<b>Object -Oriented Programming Overview</b>	<b>7 Lectures</b>
----------------	--	-------------------

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.

<b>Unit 4:</b>	<b>Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata</b>	<b>10 Lectures</b>
----------------	---	--------------------

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.

<b>Unit 5:</b>	<b>Exception Handling, Threading, Networking and Database Connectivity</b>	<b>6 Lectures</b>
----------------	--	-------------------

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.

<b>Unit 6:</b>	<b>Applets and Event Handling</b>	<b>10 Lectures</b>
----------------	-----------------------------------	--------------------

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, Gilad Bracha, 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, Gary Cornell, "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.

<b>Course-MAJOR Paper:</b>	<b>Paper Code-MAJ2L Java Programming (Lab)</b>	<b>Credits-1</b>	<b>Lab hours/Week-2</b>
----------------------------	--	------------------	-------------------------

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

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

### 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**

<b>Unit 1:</b>	<b>Introduction</b>	<b>5 Lectures</b>
Introduction, Computer Security, Threats, Harm, Vulnerabilities, Authentication Mechanisms - Passwords, Biometrics, Hardware Tokens, Authorization and Access Control Lists (ACLs).		
<b>Unit 2:</b>	<b>Firewalls</b>	<b>5 Lectures</b>
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.		
<b>Unit 3:</b>	<b>Introduction to Cyber Crime, law and Investigation</b>	<b>10 Lectures</b>
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.		
<b>Unit 4:</b>	<b>Intrusion Detection Systems (IDS)</b>	<b>10 Lectures</b>
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

<b>Course-SEC</b>	<b>Paper Code-SEC1T E1</b>	<b>Credits-1</b>	<b>Tut. hours/Week-1</b>
<b>Paper:</b>	<b>Cyber Security (Tutorial)</b>		
Cyber Security Tutorial as assigned and advised by teacher(s).			

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

***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**

<b>Unit 1:</b>	<b>Creating and Managing Presentations</b>	<b>10 Lectures</b>
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.		

<b>Unit 2:</b>	<b>Tables, Charts, SmartArt, and Media</b>	<b>10 Lectures</b>
Insert and Format Tables: Insert and Format Charts, Insert and Format SmartArt graphics, Insert and Manage Media.		

<b>Unit 3:</b>	<b>Transitions and Animations</b>	<b>10 Lectures</b>
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 Rutkowski ,Nita Rutkosky



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

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.