KOLHAN UNIVERSITY

Chaibasa, Jharkhand, India

Syllabus for
Four Year Undergraduate Programme (FYUGP)
of

Bachelor of Computer Application (BCA)

Major Courses

With Effect From
Academic Year 2022 - 2023

As Per Revised Curriculum and Credit Framework for the FYUGP under the provisions of NEP - 2020

Table of Content

Programme Structure for
Four Year Undergraduate Programme (FYUGP)
of
Bachelor of Computer Application (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
I	AEC–1	Language and Communication Skills: Hindi Composition		2	
	VAC–1	Value Added Course – 1 *(Two papers are to be selected by the students from the list of available options)*		2+2=4	
	SEC–1	Digital Education		3	
	MDC–1	Multi-Disciplinary Course – 1 *(To be selected by the students from the list of available options)*		3	
	MN–1A	Minor From Discipline–1 *(To be selected by the students from the list of available options)*		4	
	MJ–1	(Th): C Programming Language	3–0–1	3	45
		(Pr): C Programming Language Lab		1	30
II	AEC–2	Language and Communication Skills: English Composition		2	
	SEC–2	Communication Skills and Personality Development		3	
	MDC–2	Multi-Disciplinary Course – 2 *(To be selected by the students from the list of available options)*		3	
	MN–2A	Minor From Vocational Studies/Discipline – 2 *(To be selected by the students from the list of available options)*		4	
	MJ–2	(Th): Object Oriented Programming with C++	3–0–1	3	45
		(Pr): C++ Programming Language Lab		1	30
	MJ–3	(Th): Data Structure using C++	3–0–1	3	45
		(Pr): Data Structure Lab		1	30
III	AEC–3	Language and Communication Skills *(To be selected by the students from the list of available options)*		2	
	SEC–3	Mathematical & Computational Thinking Analysis		3	
	MDC–3	Multi-Disciplinary Course– 3 *(To be selected by the students from the list of available options)*		3	
	MN–1B	Minor From Discipline–1 *(To be selected by the students from the list of available options)*		4	
	MJ–4	(Th): Relational Database Management System	3–0–1	3	45

		(Pr): RDBMS (SQL) Lab		1	30
	MJ–5	(Th): Java Programming Language-I	3–0–1	3	45
		(Pr): Java Programming Language-I Lab		1	30
IV	AEC-4	Language and Communication Skills *(To be selected by the students from the list of available options)*		2	
	VAC–2	Value Added Course – 2 *(To be selected by the students from the list of available options)*		2	
	MN–2B	Minor From Vocational Studies/Discipline – 2 *(To be selected by the students from the list of available options)*		4	
	MJ–6	(Th): Java Programming Language-II	3–0–1	3	45
		(Pr): Java Programming language-II Lab		1	30
	MJ–7	(Th): Operating System-I	3–0–1	3	45
		(Pr): Operating System-I Lab		1	30
	MJ–8	(Th): Software Engineering	3–1–0	4	60
V	MN–1C	Minor From Discipline–1 *(To be selected by the students from the list of available options)*		4	
	MJ–9	(Th): Operating System-II and Introduction to Linux	3–0–1	3	45
		(Pr): Operating System-II Lab		1	30
	MJ–10	(Th): Digital Logic Design	3–1–0	4	60
	MJ–11	(Th): Web Technologies	3–0–1	3	45
		(Pr): Web Technologies Lab		1	30
	IAP	Internship/Apprenticeship/Field Work/Dissertation/Project	0–0–4	4	120
VI	MN–2C	Minor From Vocational Studies/Discipline – 2 *(To be selected by the students from the list of available options)*		4	
	MJ–12	(Th): Python Programming Language	3–0–1	3	45
		(Pr): Python Programming Language Lab		1	30
	MJ–13	(Th): Data Communication and Computer Network-I	3–1–0	4	60
	MJ–14	(Th): Computer Organization and Architecture	3–1–0	4	60
	MJ–15	(Th): Web Development using JSP	3–0–1	3	45

		(Pr): JSP Lab		1	30
VII	MN–1D	Minor From Discipline–1 *(To be selected by the students from the list of available options)*		4	
	MJ–16	(Th): Data Communication and Computer Network-II	3–1–0	4	60
	MJ–17	(Th): Computer Oriented Numerical Methods	3–1–0	4	60
	MJ–18	(Th): Computer Graphics	3–0–1	3	45
		(Pr): Computer Graphics Lab		1	30
	MJ–19	(Th): Information Security	3–0–1	3	45
		(Pr): Information Security Lab		1	30
VIII	MN–2D	Minor From Vocational Studies/Discipline – 2 *(To be selected by the students from the list of available options)*		4	
	MJ–20	(Th): Introduction to Data Science	3–0–1	3	45
		(Pr): Data Science Lab		1	30
	RC	Research Internship/Field Work/Dissertation		12	
	OR				
	AMJ–1	(Th): Artificial Intelligence	3–1–0	4	60
	AMJ–2	(Th): Assembly Language Programming	3–1–0	4	60
	AMJ–3	(Pr): Artificial Intelligence and Assembly Language Programming Lab	0–0–4	4	120
Total Credits				**160**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **AEC** (Ability Enhancement Course), **VAC**(Value Added Course), **SEC** (Skill Enhancement Course),**MDC** (Multi Disciplinary Course),**MN-1** (Minor From Discipline-1), **MN-2** (Minor From Vocational Studies/ Discipline-2), **IAP** (Internship/ Apprenticeship/ Project), **MJ** (Major Disciplinary/Interdisciplinary Courses), **AMJ** (Advance Major Disciplinary/ Interdisciplinary Courses), **RC** (Research Courses - Research Internship/Field Work/Dissertation)

Programme Objectives, Outcomes and Programme Specific Outcomes of Bachelor of Computer Application (BCA)

Programme Objectives:

1. **Foundational Knowledge in Programming and Development:**
    - Equip students with a strong foundation in programming languages such as C, C++, Java, Python, and Assembly Language, enabling them to develop efficient algorithms and software solutions.
    - Foster an understanding of Object-Oriented Programming and Data Structures for solving real-world computational problems.
2. **Mastery of Database and Data Management Concepts:**
    - Provide comprehensive knowledge of Relational Database Management Systems (RDBMS) and SQL for effective data organization, retrieval, and management.
3. **Proficiency in Software Engineering and System Analysis:**
    - Train students in Software Development Life Cycle (SDLC) methodologies, project management, and system analysis for designing scalable and maintainable software solutions.
4. **Understanding of Computer Systems and Networks:**
    - Develop a thorough understanding of Operating Systems, Computer Architecture, Digital Logic Design, and Data Communication Networks to manage system resources and ensure robust communication infrastructure.
5. **Web and Application Development Skills:**
    - Enable students to build dynamic web applications using technologies such as JSP, Web Technologies, and Python.
6. **Focus on Emerging Technologies:**
    - Introduce students to cutting-edge fields such as Data Science, Artificial Intelligence, and Information Security and preparing them for future technological advancements.
7. **Enhancement of Analytical and Problem-Solving Skills:**
    - Strengthen computational thinking and problem-solving abilities through courses in Mathematical and Computational Analysis and Computer Oriented Numerical Analysis.
8. **Interdisciplinary Learning and Communication Skills:**
    - Promote interdisciplinary knowledge through Multi-Disciplinary Courses and enhance communication skills with Language and Communication Skills modules.

Programme Outcomes:

1. Acquire knowledge of Computer application theory and algorithm principles in the design and modeling of computer based system.

2. Understand the computing concepts and their applications using the acquired board based knowledge.

3. To provide thorough understanding of nature, scope and application of computer and computer languages.

4. Identify and analyze software application problems in multiple aspect including coding, testing and implementation in industrial applications.

5. The program prepares the young professional for a range of computer applications, computer organization, and techniques of Computer Networking, Software Engineering, Web development, Database management and Advance Java.

Programme Specific Outcomes:

1. To pursue further studies to get specialization in Computer Science and Applications, Economics, Mathematics, Business Administration.

2. To pursue the career in corporate sector can opt for MBA, MCA, etc.

3. To Work in the IT sector as programmer, system engineer, software tester, junior programmer, web developer, system administrator, software developer, etc.

4. To work in public sector undertakings and Government organizations.

5. Ability to understand the changes or future trends in the field of computer application.

6. Encouraging students to convert their start-up idea to reality by implementing.

7. Students will able to understand, analyze and develop computer programs in the areas related to algorithm, system software, web design and networking for efficient design of computer–based system.

Prepared and Proposed by:

Mr. Riki Sutradhar	**Mr. Subodh Kumar**
Assistant Professor (Computer Application) Jamshedpur Co-operative College, Jamshedpur	Assistant Professor (Computer Application) Jamshedpur Co-operative College, Jamshedpur

Semester – I
Programme Structure for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
I	AEC–1	Language and Communication Skills: Hindi Composition		2	
	VAC–1	Value Added Course – 1 *(Two papers are to be selected by the students from the list of available options)*		2+2=4	
	SEC–1	Digital Education		3	
	MDC–1	Multi-Disciplinary Course – 1 *(To be selected by the students from the list of available options)*		3	
	MN–1A	Minor From Discipline–1 *(To be selected by the students from the list of available options)*		4	
	MJ–1	(Th): C Programming Language	3–0–1	3	45
		(Pr): C Programming Language Lab		1	30
			Total Credits	**20**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **AEC** (Ability Enhancement Course), **VAC** (Value Added Course), **SEC** (Skill Enhancement Course), **MDC** (Multi Disciplinary Course), **MN–1** (Minor From Discipline–1), **MN–2** (Minor From Vocational Studies/Discipline–2), **IAP** (Internship/Apprenticeship/Project), **MJ** (Major Disciplinary/Interdisciplinary Courses)

MJ–1 (Th): C Programming Language

3 Credits | 45 Minimum Class Hours | Semester I

Objectives:

This course helps the students in understanding a powerful, portable and flexible structured programming language which is suitable for both systems and applications programming. It is a robust language which contains a rich set of built–in functions and operators to write any complex program.

Learning Outcomes:

After completion of this course, a student will be able to–

- Understand and use the process of abstraction using a programming language such as ‘C’.
- Analyze step by step and develop a program to solve real world problems.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination ● Experiments – 15 Marks ● Viva-Voce – 05 Marks ● Practical File – 05 Marks														

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -‘B’		
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions	1 × 5	5
	Group -‘B’		
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	C Fundamentals	08
II	Control Structures and C Preprocessor	10
III	Arrays, Strings, Pointers, and Functions	17
IV	Structures and File Handling	10
	Total	**45**

Detailed Syllabus

Unit I: C Fundamentals (08 Hours)

History, Structures of ‘C’ Programming, Function as Building Blocks, Character Set, Tokens, Keywords, Identifiers, Variables, Constant, Data Types, Comments.

***Operators*:** Types of Operators, Precedence and Associativity, Expression, Statement and Types of Statements.

Built-in Functions: Console I/O Functions, Mathematical Functions, and Character Functions.

Unit II: Control Structures and C Preprocessor (10 Hours)

***Control Structures*:** Sequence Structure, Selection Structure (if Statement, if–else Statement, if-else-if Statement, Nested if–else Statement, switch–case Statement), Loop Structure (while, do–while, for Loop, Nested Loop), Other Statements (break, continue, goto).

***C Preprocessor*:** Types of C Preprocessor Directives.

Unit III: Arrays, Strings, Pointers, and Functions (17 Hours)

***Arrays*:** One–Dimensional Arrays (Definition, Declaration, Initialization, Accessing and Displaying Array Elements, Passing Array to a Function), Two–Dimensional Arrays (Definition, Declaration, Initialization, Accessing and Displaying Array Elements).

***Strings*:** Definition, Declaration, Initialization, Standard Library Functions.

***Pointers*:** Definition, Declaration, Initialization, Indirection Operator, Address of Operator, Operations on Pointers, Array of Pointers, Dynamic Memory Allocation.

***Functions*:** Declaration and Definition, Function Call, Types of Function, Parameter Passing (Call by Value, Call by Reference), Scope of Variables, Storage Classes (Automatic, Register, Extern, Static Variables), Recursive Function.

Unit IV: Structures and File Handling (10 Hours)

***Structures*:** Definition, Declaration, Initialization, Accessing Structure Elements, Array of Structures, Pointers and Structures, Passing Structures to a Function.

***File Handling*:** Introduction, Defining and Opening a File, Closing a File, Input/Output Operations on Text and Binary Files, Error Handling During I/O Operation, Random Access to Files.

Recommended Books:

- Rajaraman V., **Computer Programming in C** (Second Edition), New Delhi: Tata McGraw-Hill Publication, 1992
- Kanetkar Y., **Let Us C** (Third Edition), New Delhi: BPB Publications, 1999
- Gottfried, B. S., **Theory and Problems of Programming with C**, New Delhi: Tata McGraw-Hill Publication, 1997
- Balaguruswamy E. **Programming in ANSI C** (Second Edition), New Delhi: Tata McGraw-Hill Publication, 1992

Further readings:

- Dennis Ritchie, **The C Programming Language**, New Delhi: Pearson Education
- Forouzah, Ceilberg Thomson, **Structured Programming Approach Using C**, Learning Publication

- Deitel & Deitel, **C How To Program,** New Delhi: Prentice Hall India, 1996
- R. B. Patel, **Fundamental of Computers and Programming in C**, Khanna Book Publishing Company PVT. LTD. Delhi, India, 1st edition, 2008

◆

MJ–1 (Pr): C Programming Language Lab

1 Credit | 30 Minimum Class Hours | Semester I

Objectives:

This course helps the students in understanding a powerful, portable and flexible structured programming language which is suitable for both systems and applications programming. It is a robust language which contains a rich set of built–in functions and operators to write any complex program.

Learning Outcomes:

After completion of this course, a student will be able to–

- Develop modular, efficient and readable C programs by hands–on experience.
- Interpret good profound knowledge in C programming language and enable them to build programs using Control Structures, Arrays, Strings, Pointers, Functions, Structures, and File Handling to solve the real world problems.
- Illustrate memory allocation to variables dynamically and perform operations on text files.

Experiment List

Unit I: C Fundamentals	
1.	Write a program to evaluate the arithmetic expression ((A + B / C * D - E) * (F – G)). Read the values of A, B, C, D, E, F, and G from the standard input device.
2.	Write a program to check whether a number is even or odd using ternary (or, conditional) operator.
3.	Write a program to find the size of int, float, double and char data type.
Unit II: Control Structures and C Preprocessor	
4.	Write a program to find the largest number among three numbers.
5.	Write a program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +, -, *, /, % and use “**switch–case**” statement).
6.	Write a program to generate Fibonacci series up to N^{th} term.
7.	Write a program to check whether a number is palindrome or not.
8.	Write a program to check whether a number is prime or not.
9.	Define a macro that receives an array and the number of elements in the array as arguments. Write a program for using this macro to print the elements of the array.
Unit III: Arrays, Strings, Pointers, and Functions	
10.	Write a program to find the sum of all elements, average of all elements, and the second largest element in a “One–Dimensional” integer array.
11.	Write a program that lets the user perform arithmetic operations on two “Two–Dimensional” integer arrays. Your program must be menu driven, allowing the user to select the operations (e.g., Press 1 for Addition and Press 2 for Multiplication).
12.	Write a program to reverse a given string and then count the number of vowels, consonants and spaces in that reversed string.

13.	Write a program that lets the user perform string operations on standard library functions. Your program must be menu driven, allowing the user to select the operations (e.g., Press 1 to demonstrate the usage of function "**strlen()**", Press 2 to demonstrate the usage of function "**strlwr()**", Press 3 to demonstrate the usage of function "**strupr()**", Press 4 to demonstrate the usage of function "**strcat()**", and so on).
14.	Write a program to concatenate two given strings and find the length of the resultant string using pointer.
15.	Write a program to read and print an integer array. The program should input total number of elements (limit) and elements in array from user. Use dynamic memory allocation to allocate (i.e., "**malloc()**" or "**calloc()**" functions) and deallocate (i.e., "**free()**" function) array memory.
16.	Write program that use function to return the greatest common divisor of two given integers.
17.	Write a program to print the transpose of a given matrix using function.
18.	Write a program to generate Fibonacci series up to Nth term using recursive function.
Unit IV: Structures and File Handling	
19.	Write a program that lets the user perform arithmetic operations on two complex numbers. Define a structure that will hold the data for a complex number. Your program must be menu driven, allowing the user to select the operations (+, -, and *) and input the complex numbers. Furthermore, your program must consist of following functions: **(i)** Function "**showChoice()**": This function shows the options to the user and explains how to enter data. **(ii)** Function "**add()**": This function accepts two complex number structures as arguments and returns a complex number structure with the sum of the two complex numbers. **(iii)** Function "**subtract()**": This function accepts two complex number structures as arguments and returns a complex number structure with the difference of the two complex numbers. **(iv)** Function "**multiply()**":This function accepts two complex number structures as arguments and returns a complex number structure with the product of the two complex numbers.
20.	Write a program to create a text file named "**MyInfo.txt**", open it, type–in some information about yourself. Read and count the number of characters in the file. And finally, copy the contents of the file to another file named "**MyInfo_Copy.txt**" and display the contents of this file.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

Semester – II
Programme Structure for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
II	AEC–2	Language and Communication Skills: English Composition		2	
	SEC–2	Communication Skills and Personality Development		3	
	MDC–2	Multi-Disciplinary Course – 2 *(To be selected by the students from the list of available options)*		3	
	MN–2A	Minor From Vocational Studies/Discipline – 2 *(To be selected by the students from the list of available options)*		4	
	MJ–2	(Th): Object Oriented Programming with C++	3–0–1	3	45
		(Pr): C++ Programming Language Lab		1	30
	MJ–3	(Th): Data Structure using C++	3–0–1	3	45
		(Pr): Data Structure Lab		1	30
			Total Credits	**20**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **AEC** (Ability Enhancement Course), **VAC** (Value Added Course), **SEC** (Skill Enhancement Course), **MDC** (Multi Disciplinary Course, **MN–2** (Minor From Vocational Studies/Discipline–2, **MJ** (Major Disciplinary/Interdisciplinary Courses).

MJ–2 (Th): Object Oriented Programming with C++

3 Credits | 45 Minimum Class Hours | Semester II

Objectives:

The main objectives of the course are as follows–

- To get a clear understanding of object-oriented concepts.
- To understand the difference between object oriented and procedure oriented programming.
- To understand object-oriented programming through C++.
- To develop real life applications using Object Oriented Programming (OOP) concepts.

Learning Outcomes:

After completion of this course, a student will be able to–

- An understanding of the principles behind the object-oriented development process.
- Competence in the use of object-oriented programming language in the development of small to medium sized application programs.

Outline of the Course

<table>
<tr><th colspan="2" rowspan="2">Minimum Class Hours</th><th colspan="2" rowspan="2">Exam Time (Hours)</th><th colspan="2" rowspan="2">Credits</th><th colspan="9">Marks</th></tr>
<tr><th colspan="2">Semester Internal</th><th colspan="2">End Semester</th><th colspan="2">Full Mark</th><th colspan="2">Pass Marks</th><th>Total Marks</th></tr>
<tr><th>Th</th><th>Pr</th><th>Th</th><th>Pr</th><th>Th</th><th>Pr</th><th>Th</th><th>Pr</th><th>Th</th><th>Pr</th><th>Th</th><th>Pr</th><th>Th</th><th>Pr</th><th>Th + Pr</th></tr>
<tr><td>45</td><td>30</td><td>3</td><td>3</td><td>3</td><td>1</td><td>15</td><td>N/A</td><td>60</td><td>25</td><td>75</td><td>25</td><td>30</td><td>10</td><td>75+25=100</td></tr>
<tr><td colspan="15"><b><u>Marks Distribution of End Semester Practical Examination</u></b>
● Experiments – 15 Marks ● Viva-Voce – 05 Marks ● Practical File – 05 Marks</td></tr>
</table>

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -‘B’		
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions	1 × 5	5
	Group -‘B’		
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Introduction to Object Oriented Programming (OOP) and C++	10
II	Arrays, Strings and Functions	08
III	Classes and Objects, Constructors and Destructors, Operator overloading	12
IV	Inheritance, Polymorphism, and File Handling	15
	Total	**45**

Detailed Syllabus

Unit I: Introduction to Object Oriented Programming (OOP) and C++ (10 Hours)

Object Oriented Programming: Introduction, OOP vs Procedure Oriented Programming, Basic concepts of OOP - Objects, Classes, Data Abstraction, Data Encapsulation, Data Hiding, Inheritance, Polymorphism, Dynamic binding and Message passing, Benefits of OOP, Applications of OOP.

Introduction to C++: Origins of C++, Structure of C++ program, Tokens, Keywords, Identifiers, Literals/ Constants, Variables, Data Types, Input/ Output in C++, iostream header file, Namespace, main() function, Operators and Expressions in C++, Implicit Conversions, Type casting, Control Structures.

Unit II: Arrays, Strings and Functions (8 Hours)

Arrays: Declaration, Initialization, One and two-dimensional arrays.

Strings: Declaration, Initialization, Input/Output of strings, String manipulation functions, std::string class in C++.

Functions: Introduction, User defined and Library functions, Function prototypes, Calling functions, Function returning values, Passing arguments, Call by reference, Call by value, Inline functions, Default arguments, Function overloading.

Unit III: Classes and Objects, Constructors and Destructors, Operator overloading (12 Hours)

Classes and Objects: Defining a class, Members of a class - Data members and Member functions, Private, Public, and Protected visibility modes, Member function definition – inside class and outside class, Declaration of objects, Accessing class members, Friend function, Arrays of Objects, Objects as Function Arguments, Difference between Structures and Classes.

Constructors and Destructors: Constructors - Characteristics, Declaration and definition, Types of constructors - Default, Parameterized, and Copy constructor, Constructor overloading, Destructors - Characteristics, Declaration and definition.

Operator Overloading: Defining Operator Overloading, Overloading of Unary and Binary Operators, Manipulation of strings using operators, Type Conversions.

Unit IV: Inheritance, Polymorphism, and File Handling (15 Hours)

Inheritance: Introduction, Base and derived classes, Types of inheritance, Virtual base classes, Abstract classes, Constructors in inheritance, Overriding base class functions.

Polymorphism: Introduction, Compile time and Runtime polymorphism, Pointers to objects, this pointer, Pointers to derived classes, Virtual functions.

Files Handling: Introduction, Stream classes for files, Opening files, File modes, Error handling in file, Detecting end of file-eof(), Sequential input and output-put() and get(), Reading and writing objects-read() and write(), Random Access files.

Recommended Books:

- Balaguruswami, E., **Object Oriented Programming in C++**, Tata McGraw Hill Pub
- Lafore R., **Object Oriented Programming using C++**, Galgotia

Further readings:

- Herbert Schildt, **C++: The Complete Reference**, McGraw Hill
- K R Venugopal, Rajkumar Buyya, T Ravishankar, **Mastering C++**, Tata McGraw Hill

* * *

MJ–2 (Pr): C++ Programming Language Lab

1 Credit | 30 Minimum Class Hours | Semester II

Objectives:

The main objectives of the course are as follows–

- To get a clear understanding of object-oriented concepts.
- To understand the difference between object oriented and procedure-oriented programming.
- To understand object-oriented programming through C++.
- To develop real life applications using Object Oriented Programming (OOP) concepts.

Learning Outcomes:

After completion of this course, a student will be able to–

- Write basic C++ program using control structures, arrays strings etc.
- Demonstrate the concepts of reusability through the use of functions.
- Write object-oriented application using OOP concepts such as encapsulation, abstraction, polymorphism, inheritance etc.
- Use stream classes to manipulate data files.

Experiment List

Unit I: Introduction to C++ & Control Structures	
1.	Write a program to find roots of a quadratic equation, $ax^2 + bx + c = 0$. Implement following conditions – • If the discriminant is positive, then display two distinct real roots. • If the discriminant is zero, then display two equal roots. If the discriminant is negative, then display two distinct complex roots.
2.	Write a program to display Fibonacci series – 0,1,1,2,3,5,8,13, ………….N
3.	Write a menu-based program to check - 1. Perfect number 2. Palindrome number 3. Prime number
Unit II: Arrays, Strings and Functions	
4.	Write a program to input N integer elements into an array and swap the largest and lowest element.
5.	Write a program to input order of two matrices and check if it satisfies the condition for addition of the matrices or not. If it satisfies the condition then find the sum of the matrices.
6.	Write a program to input a string, find it's reverse and check whether it is palindrome or not without using any string library functions.
7.	Write a function *calculate()* which can take maximum three integer arguments x, y, and z, and returns – • x^2 if only x is passed. • x^y, if both x and y are passed. • $x^y + x^z$, if all three arguments are passed.

8.	Write a program to show difference between call by value and call by reference.
Unit III: Classes and Objects, Constructors and Destructors, Operator overloading	
9.	Write an object-oriented program to create simple calculator to perform addition, subtraction, multiplication, and division of two numbers. Define appropriate data members and member functions. Your program must use constructor and destructor.
10.	Define a class *Number* having a private integer data member *num* and member functions *input*() and *display*() to take the input and display the data respectively. Write a program using a friend function to use the class *Number* to find the sum of two numbers.
11.	Write an object-oriented program to calculate area of square, rectangle and triangle using constructor overloading.
12.	Define a class Tour with following description: Private Members: travel_id (integer) no_of_adults (integer) no_of_kids (integer) source (string) destination (string) distance (float) total_fare (float) Public Members: • A constructor to assign initial value as 0 to all integer and float data members and text "NULL" to all the string data members. • calc_fare(): To calculate total_fare as follows – For each adult: Fare (Rs.) For distance (in K.M) --- 500 >=500 250 <500 and >=300 150 <300 *For each kid the above fare will be 50% of the fare mentioned above*. • read_data(): To input the values of the data members except total_fare and call calc_fare() function. • show_data(): displays the content of all the data members for a Tour.
13.	Write a program to overload binary Addition (+) operator to find the sum of two complex numbers.
14.	Write a program to overload relational operator == to compare two strings.
15.	Write a program to demonstrate conversion from Basic to Class type.
Unit IV: Inheritance and Polymorphism	
16.	Create a base class called ***shape***. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called ***triangle*** and ***rectangle*** from the base ***shape***. Add to the base class, a member function ***get_data()*** to initialize base class data members and another member function ***display_area()*** to compute and display the area of figures. Make ***display_area()*** as a virtual function and redefine this function in the

	derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of triangles.
Unit V: Files and Streams	
17.	Write a complete employee management file handling program which will perform following operations on binary files - insert records, display record, search records and delete records.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

◆

MJ–3 (Th): Data Structure using C++

3 Credits | 45 Minimum Class Hours | Semester II

Objectives:

Data Structure is considered as one of the fundamental papers towards a more comprehensive understanding of programming and application development. Students are expected to work towards a sound theoretical understanding of Data Structures and also compliment the same with hands on experience with implementation.

The main objectives of the course are as follows–

- To understand the basic concepts of data structures and algorithms.
- To be able to practically implement the data structures like stack, queue, array, lists, tree etc.
- To understand and implement different searching and sorting techniques.

Learning Outcomes:

At the end of the course, students will be able to–

- Understand the need for Data Structures when building application.
- Able to walk through insert and delete for different data structures.
- Ability to calculate and measure efficiency of code.

Outline of the Course

Minimum Class Hours		**Exam Time (Hours)**		**Credits**		**Marks**								
						Semester Internal		**End Semester**		**Full Mark**		**Pass Marks**		**Total Marks**
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination														
● Experiments – 15 Marks					● Viva-Voce – 05 Marks					● Practical File – 05 Marks				

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	**Questions**		**Marks**
Group -'A'			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -'B'			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	**Questions**		**Marks**
Group -'A'			
1	Five very short answer type questions	1 × 5	5
Group -'B'			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Introduction to Data Structure	05
II	Arrays, Searching and Sorting	08
III	Linked list	10
IV	Stacks & Queues	12
V	Trees	10
	Total	**45**

Detailed Syllabus

Unit I: Introduction to Data Structure (05 Hours)

Basic concepts and terminology, Importance and applications of data structures, Types of data structures, Overview of operations on data structures: insertion, deletion, traversal, searching, and sorting, Performance Analysis, Performance Measurement.

Unit II: Arrays, Searching and Sorting (08 Hours)

One-dimensional arrays, Multi-dimensional arrays, Array operations and algorithms: Sorting - Selection sort, Bubble sort, Quick sort, Insertion sort, Merge Sort; Searching- Linear Search and Binary Search.

Unit III: Linked list (10 Hours)

Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly liked list, Circular linked list; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion;

Unit IV: Stacks & Queues (12 Hours)

Stacks: Basic Concepts – Definition and Representation of stacks; Array Implementation of Stack, Operations on stacks; Linked list implementation of Stacks. Applications of stacks – Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack,

Queues: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues (Deque), Priority queues; Operations on Simple queues; Linked list implementation of Queue; Applications of queue.

Unit V: Trees (10 Hours)

Introduction, Terminology, Representation of Trees, Binary Trees, Properties of Binary Trees, Binary Tree Representations, Binary Tree Traversals, Binary Search Trees: Introduction, Searching a Binary Search Tree, Inserting an Element, Deleting an Element.

Recommended Books:

- Ellis Horowitz and Sartaj Sahni; **Fundamentals of Data Structures in C;** Universities Press
- Reema Thareja; **Data Structures Using C**; Oxford University Press, India

Further Readings:

- Kamathane; **Introduction to Data structures**; Pearson Education
- Y. Kanitkar; **Data Structures Using C**; BPB

MJ–3 (Pr): Data Structure Lab

1 Credits | 30 Minimum Class Hours | Semester II

Objectives:

The main objectives of the course are as follows–

- To be able to practically implement the data structures like stack, queue, linked-list etc.
- To implement linear and non-linear data structures.
- To understand the different operations of binary search trees.
- To get familiarized to sorting and searching algorithms.

Learning Outcomes:

After completion of this course, a student will be able to–

- Understand the need for Data Structures when building application.
- Write functions to implement linear and non-linear data structure operations.
- Suggest appropriate linear and non-linear data structure operations for solving a given problem.
- Analyze various sorting methods.

Experiment List

Unit I: Searching and Sorting	
1.	Write a function to search an element in the array using binary search.
2.	Write a program to define a function to sort an array using bubble sort or selection sort or quick sort in ascending/ descending order.
Unit II: Stack	
3.	Write a menu-based program to implement push, pop and traversal operations on a stack using array or linked list.
Unit III: Queue	
4.	Write a menu-based program to implement insert, delete and traversal operations on a queue using array or linked list.
Unit IV: Linked List	
5.	Write a menu driven program that uses functions to perform the following operations on singly linked list (i) Creation (ii) Insertion (iii) Deletion (iv) Traversal.
Unit V: Tree	
6.	Write a program to create a binary search tree and perform Insertion and different types of traversals.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

Semester – III
Programme Structure for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
III	AEC–3	Language and Communication Skills *(To be selected by the students from the list of available options)*		2	
	SEC–3	Mathematical & Computational Thinking Analysis		3	
	MDC–3	Multi-Disciplinary Course– 3 *(To be selected by the students from the list of available options)*		3	
	MN–1B	Minor From Discipline–1 *(To be selected by the students from the list of available options)*		4	
	MJ–4	(Th): Relational Database Management System	3–0–1	3	45
		(Pr): RDBMS (SQL) Lab		1	30
	MJ–5	(Th): Java Programming Language-I	3–0–1	3	45
		(Pr): Java Programming Language-I Lab		1	30
			Total Credits	**20**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **AEC** (Ability Enhancement Course), **VAC** (Value Added Course), **SEC** (Skill Enhancement Course), **MDC** (Multi Disciplinary Course, **MN–2** (Minor From Vocational Studies/Discipline–2, **MJ** (Major Disciplinary/Interdisciplinary Courses).

MJ–4 (Th): Relational Database Management System

3 Credits | 45 Minimum Contact Hour | Semester III

Objectives:

The objective of the course is to provide an introduction to database management systems, with an emphasis on how to organize, maintain and retrieve efficiently, and effectively information from a DBMS. It also exposes the students to advanced database concepts.

The main objectives of the course are as follows:

- Provide an introduction to the management of database systems.
- Understand the fundamentals of relational systems including data models, database architectures, and database manipulations**.**
- To know about the database requirements and determine the entities involved in the system and their relationship to one another.
- To know about manipulation of a database using SQL
- Understand Normalization techniques.

Learning Outcomes:

At the end of the course, students will be able to:

- Describe the fundamental elements of Relational Database Management Systems
- Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra, and SQL.
- Design ER-models to represent simple database application scenarios.
- Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data.
- Improve the database design by normalization.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination														
● Experiments – 15 Marks					● Viva-Voce – 05 Marks					● Practical File – 05 Marks				

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -‘B’			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions	Marks
	Group -'A'	
1	Five very short answer type questions 1 × 5	5
	Group -'B'	
3	Descriptive type question – one out of two 5 × 1	5
4	Attendance/ overall class performance in the semester	5
	Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Introduction to Databases	10
II	Database Design and Schema	10
III	Structured Query Language (SQL)	15
IV	Database Querying and Optimization	10
	Total	**45**

Detailed Syllabus

Unit I: Introduction to Databases (10 Hours)

Introduction to Database and Database Users, Overview of Database Management Systems, Relational Database Concepts and Components, Relational Model and Relational Algebra, Relational Database Management System Architecture.

Unit II: Database Design and Schema (10 Hours)

Entity-relationship (ER) Modeling, Relational Schema Design, Functional Dependencies and Normalization (1NF, 2NF, 3NF, BCNF, 4NF), Database Constraints and Integrity.

Unit III: Structured Query Language (SQL) (15 Hours)

Introduction to SQL, SQL Data Types, Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), Transaction Control Language (TCL), Views, Indexes, and Stored Procedures, Scalar (Non-Aggregate) SQL Functions - String, Numeric and Date functions.

Unit IV: Database Querying and Optimization (10 Hours)

Basic and Advanced SQL Queries, Joins and Subqueries, Aggregate Functions and Grouping, Query Optimization Techniques.

Recommended Books:

- Abraham Silberschatz, Henry F. Korth, S. Sudharshan; **Database System Concepts** (7th Edition); Tata McGraw Hill, 2019
- Elmasri and Navathe; **Fundamentals of Database Systems** (7th Edition); Addison – Wesley, 2016

Further Readings:

- C.J. Date, A. Kannan, S. Swamynatham; **An Introduction to Database Systems** (8th Edition); Pearson education, 2009
- Raghu Ramakrishnan and Johannes Gehrke; **Database Management Systems** (3rd Edition); McGraw-Hill, 2003
- Ivan Bayross; **PL/SQL Programming**; BPB
- SQL and PL/SQL Tutorial – **https://www.w3schools.com/sql/, http://www.plsqltutorial.com/**

MJ–4 (Pr): RDBMS (SQL) Lab

1 Credit | 30 Minimum Class Hours | Semester III

Objectives:

The main objectives of the course are as follows–

- To know about the database requirements and determine the entities involved in the system and their relationship to one another.
- To know about the manipulation of database using SQL commands.

Learning Outcomes:

After completion of this course, a student will be able to–

.

- Gain knowledge on how to use SQL for Creating, Modifying and Accessing tables in Database.
- Implement Order by and Group by clauses.

Experiment List

Unit I: *CREATE, ALTER* and *DROP* Statements	
1.	Create a table **Employee** with the following fields: (Employee_Id, First_Name, Last_Name, Hire_Date, Job_Id, Salary, Manager_Id, Department_Id) Use appropriate data type and perform following task- (a) Add a new field ‘Address Char(10)’. (b) Modify the size of Address column to 20. (c) Insert any 5 records into the table. (d) Insert a record in Employee_Id, First_Name and Salary field only. (e) Display the structure of Employee table. (f) List out details of all employees. (g) Remove the field ‘Address’ from the table. (h) Copy Employee table to Emp_backup. (i) Remove the table Employee from the database. Change the name of the table from Emp_backup to Employee.
Unit II: *UPDATE* and *DELETE* Statements	
2.	Create an Emp table with the following fields: (EmpNo, EmpName, Job, Basic, DA, HRA, PF, GrossPay, NetPay) (Calculate DA as 30% of Basic and HRA as 40% of Basic and PF as 12.5% of Basic) (a) Insert Five Records in the following fields (EmpNo, EmpName, Job, Basic) (b) Calculate DA, HRA, PF, GrossPay (Basic+DA+HRA) and NetPay (GrossPay-PF) of all employees. (c) Display all records. (d) If NetPay is less than <Rs. 10,000 add Rs. 1200 as special allowances. (e) Delete all ‘Clerks’ having Basic 5000 or less.
Unit III: Integrity Constrains	

<table>
<tr><td>3.</td><td>Create a table named Library with appropriate data type of following structure:
(Book_id, Title, Author, Subject, Publisher, Quantity, Price, Student_id)
Apply following constraints on the field
(i) Book_id must be Primary Key
(ii) Title must be Unique
(iii) Quantity should be more than 100
(iv) Price should be between Rs. 10 and Rs. 5000

(a) View all the constraints from the data dictionary
(b) Add Foreign Key constraints to Student_id column which references to Student(Student_id). [Create Student(Roll, Name, Book_id(PK)) before adding the Foreign Key constraints]
(c) Describe the structure of the table.
(d) Insert records to verify the constraints.</td></tr>
<tr><td colspan="2"><b>Unit V: <i>SELECT</i> Statement</b></td></tr>
<tr><td>4.</td><td>A company wishes to maintain a database to automate its operations. Company is divided into certain departments and each department consists of employees. The following two tables describes the automation schemas: Dept (deptno, dname, loc)
Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)
(a) Create above tables with appropriate data types
(b) Insert details of three departments and details of 5 employees.
(c) List the employee’s name and salary, whose experience is greater than 10 years.
(d) Display unique jobs from the table.
(e) Display employees of department no. 20 and 30 who have salary between 20000 and 30000.</td></tr>
<tr><td>5.</td><td>Consider the table
Dept (deptno, dname, loc)
Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)
created earlier, write following query:
(a) To list all employee who do not have commission.
(b) To list all ‘Salesman’ of dept. 30 who have commission.
(c) To list employee name and annual salary (sal*12).
(d) To list the name of all employees whose name begins with letter ‘Raj’.
(e) Display empno, ename, deptno and dname of all employee.
(f) List the name of those employees who earn more than all of the employees of ‘Sales’ dept.</td></tr>
<tr><td colspan="2"><b>Unit VI: <i>ORDER BY</i> and <i>GROUP BY</i> Clause, Aggregate Functions</b></td></tr>
<tr><td>6.</td><td>Create a table Library with appropriate data type of following structure:
(Book_id, Title, Author, Subject, Publisher, Quantity, Price)

(a) Insert any ten records
(b) Calculate total quantity of books of each subject.
(c) Calculate average price of books of each publisher.
(d) Display total quantity, maximum and minimum price of subjects ‘Java’ and ‘Python’.
(e) Display list of all the books with price more than 300 in ascending order of price.
(f) List Author wise count of books in alphabetical order.</td></tr>
<tr><td colspan="2"><b>Unit VII: Join and Sub-Query</b></td></tr>
</table>

7.	Consider the table Dept (deptno, dname, loc) Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno) created earlier, write following query: (a) Display the manager who is having maximum number of employees working under him? (b) List the names of employees, who take highest salary in their departments. (c) Create a view Emp_Dept, which contains Employee name, job, salary and department name.
Unit IV: *ROLLBACK, COMMIT, GRANT* and *REVOKE* Statement	
8.	Create Teacher table with the following fields (Name, DeptNo, Date_of_joining, DeptName, Location, Salary) (a) Insert five records. (b) Give Increment of 25% salary for Mathematics Department. (c) Perform Rollback command. (d) Give Increment of 15% salary for Commerce Department. (e) Perform commit command.
9.	(a) Create a new user ‘ku’ having password ‘ku123’ (b) Grant all privileges to the user ‘ku’ (c) Create a table BCA(adm_id, name) (d) Revoke all privileges from the user ‘ku’ (e) Grant only CREATE and SELECT privilege from user ‘ku’ on table BCA (f) Revoke SELECT privilege from user ‘ku’ from table BCA.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

MJ–5 (Th): Java Programming Language-I

3 Credits | 45 Minimum Class Hours | Semester III

Objectives:

The Java Programming Language course is designed to provide students with a comprehensive understanding of the Java programming language and its application in software development. The course aims to equip students with the necessary skills to design, implement, and debug Java programs. Students will learn the fundamental concepts of object-oriented programming and gain hands-on experience in developing Java applications. The main objectives of the course are as follows–

- Understand the basic concepts of Java programming language.
- Design and implement Java programs using object-oriented principles.
- Apply control structures, data types, and operators in Java programming.
- Use Arrays for data storage and manipulation.
- Apply packages to organize the group of classes, interfaces etc.

Learning Outcomes:

By the end of the course, students will be able to:

- Design and implement Java programs that demonstrate a clear understanding of object-oriented programming principles.
- Apply control structures, data types, and operators effectively in Java programming to solve problems.
- Utilize arrays for efficient data storage, retrieval, and manipulation.
- Implement packages to organize classes and interfaces into a single unit.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100

Marks Distribution of End Semester Practical Examination

- Experiments – 15 Marks
- Viva-Voce – 05 Marks
- Practical File – 05 Marks

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -‘B’		
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions	1 × 5	5
	Group -'B'		
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Introduction and Fundamentals of Java	15
II	Arrays and Strings	10
III	Class, Objects and Methods	15
IV	Packages	05
	Total	**45**

Detailed Syllabus

Unit I: Introduction and Fundamentals of Java (15 Hours)

Introduction to Java and its features, Java Runtime Environment, Java Virtual Machine, Java Development Kit, Java Program Structure, Tokens- Keywords, Identifiers, Constants, Primitive and reference data types in Java, Variable declaration & initialization, Types of variables such as local, instance, and static variables, Input/ Output in Java, Java Operators and Expressions, Operator Precedence and Associativity, Type Conversion in Expressions, Mathematical Functions, Wrapper classes.

Control Structures: Decision-making, Branching and Looping statements.

Unit II: Arrays and Strings (10 Hours)

Array: Introduction, One Dimensional Arrays, Declaration, Creation, Initialization of Arrays, Two Dimensional Arrays.

String: Fundamentals of Characters and Strings, The String Class and its methods, String Operations.

Unit III: Class, Objects and Methods (15 Hours)

Introduction, Defining a Class, Fields declaration, Method declaration, Creating object, Accessing class members, Access or Visibility Modifier, this keyword, Method overloading, Constructors, Constructor overloading, Garbage Collection, The Finalize ( ) Method.

Inheritance and Polymorphism: Inheritance Basics, Super and Sub class, Types of Inheritance, Overriding methods, super keyword, final keyword, Static and Dynamic Binding, Abstract methods and classes, Compile-time Polymorphism and Runtime Polymorphism.

Interfaces: Defining Interfaces, Implementing Interfaces, Extending Interfaces, Accessing Interface variable, Multiple Inheritance through Interfaces.

Unit IV: Packages **(5 Hours)**

Introduction, Java API Packages, Naming Convention, Creating Packages, Accessing a Package, Using a Package, Adding a Class to a Package, Hiding Classes.

Recommended Books:

- E. Balagurusamy; **Programming with JAVA**; McGraw Hill, New Delhi
- Joel Murach, Michael Urban; **Murach's Beginning Java with Net Beans**; SPD

Further readings:

- Herbert Schildt; **Java:The Complete Reference**; McGraw Hill
- Raj Kumar Buyya; **Object Oriented Programming with JAVA**; McGraw Hill
- Ken Arnold, James Gosling; **The Java Programming Language**; Addison Wisely
- Wiley; **Java 6 Programming Black Book**; Kogent Learning Solutions

* * *

MJ–5 (Pr): Java Programming Language-I Lab

1 Credit | 30 Minimum Class Hours | Semester III

Objectives:

The main objectives of the course are as follows–

- To teach the students basics of JAVA programs and its execution.
- Use Arrays for data storage and manipulation.
- To organize classes and interfaces in to a single unit using packages.
- To make the students learn concepts Object Oriented Programming.

Learning Outcomes:

After completion of this course, a student will be able to–

.

- Use Java compiler and other platform to write and execute java program.
- Utilize arrays for efficient data storage, retrieval, and manipulation.
- Understand and Apply Object oriented features and Java concepts.

Experiment List

Unit I: Data Types, Operators and Expressions, Selection and Loop Statements	
1.	Write a program to input and display different types of data values using Scanner or Stream class.
2.	Write a program to create a simple calculator which can perform basic arithmetic operations like addition, subtraction, multiplication or division, exponent (x^y), and square root depending upon the user input.
3.	Write a program to convert primitive to wrapper class and wrapper to primitive.
4.	Write a program to input length of three sides of a triangle. Then check if these sides will form a triangle or not. If sides form a triangle, then display the type of the triangle with its area and perimeter.
5.	Write a program to find factorial of a number. Input the number as command line argument.
6.	Write a menu-based program to check Prime, Armstrong, and Perfect number.
7.	Write a menu-based program to convert following – • Decimal to Binary Number • Binary to Decimal Number
Unit II: Arrays, Strings	
8.	Write a java program to check whether the elements of an array are sorted or not. If sorted, display the order, otherwise sort elements in ascending order or descending order as per user's choice.
9.	Write a program to input order of two matrices and check if it satisfies the condition for product of the matrices or not. If it satisfies the condition then find the product of the matrices. *(Hint: Two matrices can be multiplied if and only if they satisfy the following condition:* *The number of columns present in the first matrix should be equal to the number of rows present in the second matrix.)*

10.	Write a program to input a string and display number of vowels and consonant in each word. For example, if the string is "Kolhan University Chaibasa", then the output will be – Kolhan: Vowels = 2, Consonants = 4 University: Vowels = 4, Consonants = 6 Chaibasa: Vowels = 4, Consonants = 4
Unit III: Class, Objects and Methods	
11.	Write a program in Java with class Rectangle with the data fields width, length, area and color. The length, width and area are of double type and color is of string type .The methods are set_ length () , set_width (), set_ color(), and find_ area (). Create two object of Rectangle and compare their area and color. If area and color both are same for the objects then display "Matching Rectangles" otherwise display "Non matching Rectangle".
12.	Create a class Account with two overloaded constructors. First constructor is used for initializing, name of account holder, account number and initial amount in account. Second constructor is used for initializing name of account holder, account number, addresses, type of account and current balance. Account class is having methods Deposit (), Withdraw (), and Get_Balance(). Make necessary assumption for data members and return types of the methods. Create objects of Account class and use them.
13.	Write a Java program to create a shape class and derive, square and circle classes from shape class. Define appropriate constructor for all the three classes. Define a method Area( ) to calculate area of circle and square in respective class. Assume PI = 3.14 and declare it as a final variable in circle class.
14.	Define an Employee class with suitable attributes having getSalary() method, which returns salary withdrawn by a particular employee. Write a class Manager which extends a class Employee, override the getSalary() method, which will return salary of manager by adding traveling _allowance, house rent allowance etc. Use default and parameterized constructors to initialize data.
15.	Write a java program which creates an interface having 2 methods add () and sub(). Create a class which implements the above interface for addition and subtraction of two numbers respectively.
16.	Write a program to demonstrate the multiple inheritance using interfaces.
Unit IV: Packages	
17.	Create and implement a package having two public classes.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

Semester – IV
Programme Structure for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
IV	AEC-4	Language and Communication Skills *(To be selected by the students from the list of available options)*		2	
	VAC–2	Value Added Course – 2 *(To be selected by the students from the list of available options)*		2	
	MN–2B	Minor From Vocational Studies/Discipline – 2 *(To be selected by the students from the list of available options)*		4	
	MJ–6	(Th): Java Programming Language-II	3–0–1	3	45
		(Pr): Java Programming language-II Lab		1	30
	MJ–7	(Th): Operating System-I	3–0–1	3	45
		(Pr): Operating System-I Lab		1	30
	MJ–8	(Th): Software Engineering	3–1–0	4	60
			Total Credits	**20**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **AEC** (Ability Enhancement Course), **VAC** (Value Added Course), **SEC** (Skill Enhancement Course), **MDC** (Multi Disciplinary Course), **MN–2** (Minor From Vocational Studies/Discipline–2), **MJ** (Major Disciplinary/Interdisciplinary Courses).

MJ–6 (Th): Java Programming Language-II

3 Credits | 45 Minimum Class Hours | Semester IV

Objectives:

The Java Programming Language course is designed to provide students with a comprehensive understanding of the Java programming language and its application in software development. The course aims to equip students with the necessary skills to design, implement, and debug Java programs. The main objectives of the course are as follows–

- Understand the concept of multithreading and its advantages and learn how to create and manage threads in Java.
- Explore different file handling techniques, such as byte and character streams.
- Explore the Java Collection Framework and its key interfaces and classes.
- Explore Java's Swing and JavaFX libraries for GUI development.
- Learn how to establish database connections and manage connections.

Learning Outcomes:

By the end of the course, students will be able to:

- Create and manage concurrent threads in Java applications.
- Perform read and write operations on text and binary files.
- Apply collection framework concepts to solve real-world problems.
- Develop interactive applications with responsive user interfaces.
- Develop robust database-driven Java applications.

Outline of the Course

Minimum Class Hours		**Exam Time (Hours)**		**Credits**		**Marks**								
						Semester Internal		**End Semester**		**Full Mark**		**Pass Marks**		**Total Marks**
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination														
● Experiments – 15 Marks					● Viva-Voce – 05 Marks					● Practical File – 05 Marks				

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -'B'		
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions	1 × 5	5
	Group -'B'		

3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Multithreading and Exception Handling	10
II	Managing Input/ Output Files in Java, Java Collection Framework	10
III	Event and GUI Programming	15
IV	JDBC	10
	Total	**45**

Detailed Syllabus

Unit I: Multithreading and Exception Handling (10 Hours)

Multithreading: Introduction, Creating Threads, Extending the Thread Class, Stopping and Blocking a Thread, Life Cycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, and Synchronization, Deadlock.

Exception Handling: Introduction, Types of Errors, Exceptions, Syntax of Exception Handling Code, Multiple Catch Statements, Using finally Statement, Throwing Our Own Exceptions.

Unit II: Managing Input/ Output Files, Java Collection Framework (10 Hours)

Managing Input/ Output Files: Introduction, Streams in Java, Stream Classes – Character (File Reader & Writer classes) & Byte Streams (File I/O Stream Classes), Using the File class, Creating Files, Reading/Writing Characters and Bytes to File, Random Access Files.

Java Collection Framework: Introduction, Collection Framework Interfaces (Set, List, Queue, Iterator) and Classes (ArrayList, Vector, LinkedList, Stack).

Unit III: Event and GUI programming (15 Hours)

Event Handling: Event Classes, Sources of Event, Event Listeners, Key and Mouse Event Handling.

AWT (Abstract Windows Toolkit) and Swings: Architecture, Component, Container, Panel, Window, Frame, Canvas, Components of AWT (Button, Label, Checkbox, CheckboxGroup, Choice, List, TextField etc.) and Swing (JLabel, JButton, JCheckBox, JRadioButton, JList, JComboBox, JTextField, JTextArea, JPanel, JFrame etc.), Working with Graphics, Working with Color and Font, Adding and Removing Controls, Responding To Controls, Layout Managers (Flow Layout, Grid Layout, Card Layout, Border Layout), Handling Events.

Unit IV: JDBC (10 Hours)

Database Connectivity using JDBC: JDBC Architecture, JDBC Drivers, Using Connection, Statement & Resultset Interfaces for Manipulating Data with Databases.

Recommended Books:

- E. Balagurusamy; **Programming with JAVA**; McGraw Hill, New Delhi
- Joel Murach, Michael Urban; **Murach's Beginning Java with Net Beans**; SPD

Further readings:

- Herbert Schildt; **Java:The Complete Reference**; McGraw Hill

- Raj Kumar Buyya; **Object Oriented Programming with JAVA**; McGraw Hill
- Ken Arnold, James Gosling; **The Java Programming Language**; Addison Wisely
- Wiley; **Java 6 Programming Black Book**; Kogent Learning Solutions

* * *

MJ–6 (Pr): Java Programming Language – II Lab

1 Credit | 30 Minimum Class Hours | Semester IV

Objectives:

The main objectives of the course are as follows–

- Understand the concept of multithreading and its advantages and learn how to create and manage threads in Java.
- Explore different file handling techniques, such as byte and character streams.
- Explore the Java Collection Framework and its key interfaces and classes.
- Explore Java's Swing and JavaFX libraries for GUI development.
- Learn how to establish database connections and manage connections.

Learning Outcomes:

After completion of this course, a student will be able to–

- Create and manage concurrent threads in Java applications.
- Perform read and write operations on text and binary files.
- Apply collection framework concepts to solve real-world problems.
- Develop interactive applications with responsive user interfaces.
- Develop robust database-driven Java applications.
- Apply Microsoft Windows and Unix/Linux operating system basic commands.

Experiment List

Unit I: Multithreading and Exception Handling	
1.	Write a program that creates 2 threads - each displaying a message (Pass the message as a parameter to the constructor). The threads should display the messages continuously till the user presses ctrl+C.
2.	Write a Java program to use the try, catch and finally block to handle the exception.
3.	Write a program to illustrate the throws keyword in Java.
4.	Create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception “AgeNotWithinRangeException”. If name contains numbers or special symbols raise exception “NameNotValidException”. Define the two exception classes.
Unit II: Managing Input/ Output Files, Java Collection Framework	
5.	Write a java program to copy the contents from one file to another file.
6.	Write a java program to read the student data from user and store it in the binary file. Also read the file and display all the records.
7.	Write a program to implement Vector class and its methods to store N numbers into a vector and display its sum and average.
8.	Write a program to add, retrieve and remove the element from the ArrayList.

9.	Write a program to implement LinkedList and perform different operations on it.
Unit III: Event and GUI programming	
10.	Display various graphics object such as rectangle, square, oval, circle, lines etc. with different colours on screen.
11.	Write a Java program to create a simple calculator with basic operations such as +, -, /, * using java swing elements. Use appropriate layout.
12.	Write a program to create a combo box which includes list of subjects and radio buttons to show different colour options. Display the selected subject in the text field with selected colour.
Unit IV: JDBC	
13.	Develop an application (GUI or Console based) to connect to a database created in MYSQL/ MS–ACCESS/ SQL–SERVER/ ORACLE and Perform basic operations of Selection, Insertion, Modification and Deletion on the database using JDBC.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

◆

MJ–7 (Th): Operating System – I

3 Credits | 45 Minimum Class Hours | Semester IV

Objectives:

The objective of this course is to introduce the students to a layer of software called the Operating Systems, whose job is to manage all the devices of a computer system and provide user programs with a simple interface to the hardware. This paper will familiarize the students with the concepts of process management, process synchronization, and the potential problem of deadlocks.

Learning Outcomes:

After completion of this course, a student will be able to–

- Understand the basic working process of an operating system.
- Understand the importance of process and scheduling.
- Understand the issues in deadlock.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination														
● Experiments – 15 Marks					● Viva-Voce – 05 Marks					● Practical File – 05 Marks				

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -'B'		
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions	1 × 5	5
	Group -'B'		
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Overview	15
II	Process Management	20
III	Process Synchronization	10
	Total	**45**

Detailed Syllabus

Unit I: Overview (15 Hours)

Introduction: Definition of Operating Systems, Functions of Operating Systems, Generations of Operating System, Types of Operating System {Batch Operating System, Mainframe Operating System, Desktop Operating System, Time Sharing (or, Multi–tasking) Operating System, Multi–processing Operating System, Real–Time Operating System, Distributed Operating System, Clustered Operating System, Network Operating System, Mobile Operating System, Embedded Operating Systems}.

Operating System Structures: Operating System Components (Kernel, Hardware Device Drivers, Disk Access and File Systems, Security, User Interface), Operating System Services, System Calls, Application Programming Interface (API), User Mode and Kernel Mode, Types of System Calls, System Programs, Operating System Design and Implementation, Operating System Structure.

Unit II: Process Management (20 Hours)

Processes: Process Concept (The Process, Process State, Process Control Block), Process Scheduling (Scheduling Queues, CPU Scheduling, Context Switch), Operations on Processes (Process Creation, Process Termination), Process Termination in Android Operating System, Interprocess Communication (Independent Process, Co–operative Process, IPC in Shared–Memory Systems, IPC in Message–Passing Systems).

Multithreaded Programming: Introduction to thread (Components of Thread, Single–Threaded Process, Multithreaded Process), Differences between Process and Thread, Types of Thread (User Thread, Kernel Thread), Multithreading Models (Many–to–One, One–to–One, Many–to–Many), Advantages and Disadvantages of Thread.

CPU Scheduling: Basic Concepts (CPU–I/O Burst Cycle, CPU Scheduler, Preemptive and Non-preemptive Scheduling, Dispatcher), Scheduling Criteria, Scheduling Algorithms (First–Come, First–Served Scheduling, Shortest–Job–First Scheduling, Priority Scheduling, Round–Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling).

Unit III: Process Synchronization (10 Hours)

Overview: Introduction to Process Synchronization, Race Condition, The Critical–Section Problem, Peterson's Solution.

Deadlocks: Definition of a Deadlock, Conditions for Deadlock, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance (Resource–Allocation–Graph Algorithm, Banker's Algorithm), Deadlock Detection, Recovery from Deadlock (Process Termination, Resource Preemption).

Recommended Books:

- Abraham Silberschatz, Peter B. Galvin, Greg Gagne; **Operating System Concepts** (Tenth Edition); New York: John Wiley and Sons, 2018
- Andrew S. Tanenbaum; **Modern Operating Systems** (Third Edition); New Delhi: Prentice-Hall India, 2002
- William Stallings; **Operating Systems** (Fourth Edition), New Delhi: Prentice-Hall India, 2003

Further readings:

- Harvey M. Deitel; **Operating Systems** (Second Edition); New Delhi: Pearson Education
- Pramod Chandra P. Bhatt; **An Introduction to Operating Systems Concept**; New Delhi: Prentice-Hall India

- Maurice J. Bach; **The Design of the Unix Operating System**; New Delhi: Prentice-Hall India, 1992
- Brian W. Kernighan, Rob Pike; **The Unix Programming Environment**, Pearson Education, 1984

◆

MJ–7 (Pr): Operating System – I Lab

1 Credits | 30 Minimum Class Hours | Semester IV

Objectives:

The main objectives of the course are as follows–

- To make students familiar with the Microsoft Windows and Linux command–line environment. This course serves as a platform for the subsequent labs related to Process Management, Process Scheduling, etc.

Learning Outcomes:

After completion of this course, a student will be able to–

- Apply Microsoft Windows and Unix/Linux operating system basic commands.
- Understand different Commands for process management in Microsoft Windows and Linux Operating System.

Experiment List

Unit I: Basic Microsoft Windows and Linux Commands	
1.	Briefly explain the following basic Microsoft Windows commands with examples: logoff, shutdown, tasklist, taskkill, chkdsk, sfc /scannow, format, diskpart, ver, systeminfo, dir, cd, copy, del, ren, ipconfig, ping, tracert, netsh wlan show profile name=WiFi_SSID key=clear, cls, help, powercfg, clip, color, compact, erase, doskey, driverquery, exit, find, and hostname.
2.	Briefly explain the following basic Linux commands with examples: man, history, pwd, who, finger, passwd, exit, logout, shutdown, mkdir, cd, ls, cat, cp, cmp, mv, paste, rm, rmdir, find, more, head, tail, echo, sed, grep, awk, date, time, cal, diff, file, sort, chmod, chown, du, and compress.
Unit II: Different Commands for Process Management in Linux OS/Microsoft Windows OS	
1.	Launch a process (or, program or, application) in the foreground from terminal/command prompt. After launching the process, stop it during the execution.
2.	Display the list of running in foreground, running in background, force stopped, and pending processes.
3.	Resume the force stopped and pending processes while keeping them running in the background.
4.	Resume the force stopped and pending processes while keeping them running in the foreground.
5.	Launch a process in the background directly.
6.	Launch a process in the background directly without getting impacted by the closing of the terminal/command prompt.
7.	Display a dynamic real–time table of processes of your Linux/Windows operating system.
8.	How do we terminate a running process on our Linux/Windows operating system?
9.	Report the used and available space in the primary memory storage in human–friendly units like megabytes or kilobytes.

10.	Report the used and available space in the secondary memory storage in human–friendly units like megabytes or kilobytes.
11.	How can you prioritize a process as per your requirement?

Note: Additional lab assignments may be included based on topics covered in the theory paper.

◆

MJ–8 (Th): Software Engineering

4 Credits | 60 Minimum Contact Hours | Semester IV

Objectives:

This course helps students to understand the software development process and design. It also helps the students to understand about the different stages of software development, various process models and software engineering principles.

The main objectives of the course are as follows–

- To provide students an in depth understanding of software engineering principles.
- To prepare the students to develop the skills necessary to handle software projects.
- To make the students aware of the importance of software engineering principles in designing software projects.
- To make students familiar with cost estimation and testing measurement in software development process.

Course Outcomes:

At the end of the course, students will be able to–

- Understand the importance of the stages in the software lifecycle.
- Understand the various process models.
- To design software by applying the software engineering principles.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Mark		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=75

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -‘B’		
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions	1 × 5	5
	Group -‘B’		
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Software Engineering Concepts, Process Model	14
II	Requirements Analysis and Specification	14
III	Software Design	12
IV	Coding and Testing of Software	12
V	Software Maintenance	8
	Total	**60**

Detailed Syllabus

Unit I: Software Engineering Concepts (14 Hours)

Introduction**:** Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Software Quality Attributes.

SDLC or Software Development Process Models: Classical Waterfall Model, Iterative Waterfall Models, Prototype Model, RAD Model, Agile Development Models, Spiral Model.

Software Project Management**:** Size Estimation- Line of Code (LOC) and Function Point (FP) Metric, Cost Estimation-Delphi and Basic COCOMO Model.

Unit II: Requirements Analysis and Specification (14 Hours)

Software Requirements Specification (SRS): SRS Documents, their Characteristics and Organization.

Requirement Engineering Process**:** Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study.

Function Oriented Modeling**:** Data Flow Diagrams (DFD), Entity Relationship Diagrams (ERD), Data Dictionaries, and Decision Tables.

Unit III: Software Design (12 Hours)

Classification, Software Design Approaches, Cohesion and Coupling, Function Oriented Software Design, Introduction to Object Oriented Design, Need of UI design, Design issues, The UI design Process.

Unit IV: Coding and Testing of Software (12 Hours)

Coding: Coding Standards and Guidelines, Software Documentation.

Testing: Unit Testing, Black Box Testing, White Box Testing, Debugging, Program Analysis Tools, System Testing.

Unit V: Software Maintenance (08 Hours)

Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Models, Estimation of Maintenance Cost.

Recommended Books:

- Rajib Mall; **Fundamentals of Software Engineering**; Prentice Hall of India, New Delhi
- Ian Sommerville; **Software Engineering**; Addison Wesley

Further Readings:

- Richard Fairley; **Software Engineering Concepts**; Tata McGraw Hill, New Delhi
- Pankaj Jalote; **Software Engineering**; Wiley

* * *

Semester – V
Programme Structure for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
V	MN–1C	Minor From Discipline–1 *(To be selected by the students from the list of available options)*		4	
	MJ–9	(Th): Operating System-II and Introduction to Linux	3–0–1	3	45
		(Pr): Operating System-II Lab		1	30
	MJ–10	(Th): Digital Logic Design	3–1–0	4	60
	MJ–11	(Th): Web Technologies	3–0–1	3	45
		(Pr): Web Technologies Lab		1	30
	IAP	Internship/Apprenticeship/Field Work/ Dissertation/ Project	0–0–4	4	120
			Total Credits	**20**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **AEC** (Ability Enhancement Course), **VAC** (Value Added Course), **SEC** (Skill Enhancement Course), **MDC** (Multi Disciplinary Course), **MN–1** (Minor From Discipline–1, **MJ** (Major Disciplinary/Interdisciplinary Courses).

MJ–9 (Th): Operating System – II and Introduction to Linux

3 Credits | 45 Minimum Class Hours | Semester V

Objectives:

The objective of this course is to introduce the students to memory management strategies, virtual memory, and file system. The students will also learn about the Linux operating system.

Learning Outcomes:

At the end of the course, students will be able to–

- Understand the need for memory management in operating systems.
- Understand how program memory addresses relate to physical memory addresses.
- Grab the idea of virtual memory management, including paging and segmentation.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination ● Experiments – 15 Marks ● Viva-Voce – 05 Marks ● Practical File – 05 Marks														

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -‘B’			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions	1 × 5	5
Group -‘B’			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Memory Management	20
II	File System	10
III	Linux Operating System	15
	Total	**45**

Detailed Syllabus

Unit I: Memory Management (20 Hours)

***Memory Management Strategies*:** Introduction to Memory Management, Main Memory, Registers, Cache Memory, Memory Protection (Base Register, Limit Register), Address Binding, Logical and Physical Address Space, Static and Dynamic Loading, Static and Dynamic Linking, Swapping, Memory Management Technique [Uniprogramming, Multiprogramming {Contiguous Memory Allocation (Fixed Size Partitioning, Variable Size Partitioning, Partitioning Algorithms, Fragmentation), Non–contiguous Memory Allocation (Paging, Segmentation, Segmentation with Paging)}].

***Virtual Memory*:** Introduction, Demand Paging, Page Fault, Basics of Page Replacement, Need for Page replacement.

Unit II: File System (10 Hours)

File Concept {File Attributes, File Operations, File Types (Regular files, Directories, Character Special Files, Block Special Files), File Extension, File Structure (Unstructured Format, Simple Record Structure, Complex Structure)}, Access Methods (Sequential, Direct), Protection (Types of Access, Access Control).

Unit III: Linux Operating System (15 Hours)

History, Design Principles, Kernel Modules, Process Management, Scheduling, Memory Management, File Systems, Input and Output, Security.

Recommended Books:

- Abraham Silberschatz, Peter B. Galvin, Greg Gagne; **Operating System Concepts** (Tenth Edition); New York: John Wiley and Sons, 2018
- Andrew S. Tanenbaum; **Modern Operating Systems** (Third Edition); New Delhi: Prentice–Hall India, 2002
- William Stallings; **Operating Systems** (Fourth Edition), New Delhi: Prentice–Hall India, 2003

Further readings:

- Harvey M. Deitel; **Operating Systems** (Second Edition); New Delhi: Pearson Education
- Pramod Chandra P. Bhatt; **An Introduction to Operating Systems Concept**; New Delhi: Prentice–Hall India
- Maurice J. Bach; **The Design of the Unix Operating System**; New Delhi: Prentice–Hall India, 1992
- Brian W. Kernighan, Rob Pike; **The Unix Programming Environment**, Pearson Education, 1984

◆

MJ–9 (Pr): Operating System-II Lab

1 Credit | 30 Minimum Class Hours | Semester V

Objectives:

The main objective of this course is to develop the skills of shell scripting. In this course, students are going to be learning about basic syntax of shell scripting.

Learning Outcomes:

After completion of this course, a student will be able to–

- Apply Unix/Linux operating system basic commands.
- Identify and use Unix/Linux utilities to organize directory structures with appropriate security.
- Develop shell scripts to perform complex tasks.
- Implement different types of process scheduling algorithms like First–Come, First–Served (FCFS) Scheduling, Shortest–Job–First (SJF) Scheduling, and Priority (P) Scheduling using shell scripts.

Experiment List

Unit I: Application of Linux Commands

1. Create a directory hierarchy including files in your home directory according to the following questions–

    (i) Create a new directory with the name *BCA* using **mkdir** command in your home directory.
    (ii) Change the current directory to *BCA* using **cd** command.
    (iii) In the directory *BCA*, create another directory, called as *MJ_10* using **mkdir** command.
    (iv) Change the current directory to *MJ_10* using **cd** command.
    (v) Create a text file *MJ_10_Info.txt* using **vi** command (or, any text editor) and write basic information about your tenth major course (i.e., Operating System-II and Introduction to Linux). Save and close it.
    (vi) Copy the contents of *MJ_10_Info.txt* to *MJ_10_Info_Copy.txt*in the same directory using **cp** command.
    (vii) Open *MJ_10_Info_Copy.txt* using **vi** command (or, any text editor) and verify that it is a copy of *MJ_10_Info.txt*. Close the file.
    (viii) Rename *MJ_10_Info_Copy.txt* as *Major_10_Info_Clone.txt* using **mv** command.
    (ix) Change the current directory to *BCA* using **cd** command and check if the file has been renamed by listing the contents of the directory *BCA*including hidden files using **ls** command.
    (x) Change the current directory to your home directory using **cd** command.
    (xi) Copy the directory hierarchy of *BCA* to *BCA_Copy* using **cp –r** command.
    (xii) Change the current directory to *BCA_Copy*using **cd** command and check the contents of the directory *BCA_Copy* using **ls** command.
    (xiii) Change the current directory to your home directory using **cd** commandand delete the whole directory *BCA_Copy* using **rm –r** command.

2. List all the files contained in your home directory including hidden files.

3. Display the sorted list of all files contained in your home directory by file size.

4. Use the **who** command and redirect the result to a file called *Logged_In_Info.txt*. Use the **more** command to see the contents of *Logged_In_Info.txt*.

5. Use the **who** and **date** commands in sequence (in one line) such that the output of **who** command will display on the screen and the output of **date** command will be redirected to a file called*Todays_Date.txt*. Use the **more** command to check the contents of *Todays_Date.txt*.

6. Write a **sed** command that swaps the first and second words in each line in a file.

7. Create a text file *CourseList.txt* using **cat** command in your home directory and write all the Course Names in Semester – V. Now count the number of characters in *CourseList.txt* using **wc** command and save the result in the file *CourseList_CharCount.txt*.

8. Download the file **https://www.nta.ac.in/Download/FAQ_TPC_STUDENTS.pdf** from the web to your current directory.

9. Do a long listing for files stored in the **/boot** directory, and include the size of each file in human–friendly units like megabytes or kilobytes.

Unit II: Programming in Shell Script

1. Write a program in Shell Script to see current date, time, username, and current directory.

2. Krishna’s basic salary is input through the keyboard. His dearness allowance is 40% of his basic salary, and house rent allowance is 18% of the basic salary. Write a program in Shell Script to calculate his gross salary.

3. The distance between Ranchi and Chaibasa (i.e., 150 kilometers approximately) by train is input through the keyboard. Write a program in Shell Script to convert and print this distance in meters, feet, inches and centimeters.

4. An integer is input through the keyboard. Write a program in Shell Script to find out whether it is an odd or even number.

5. If a five digit number is input through the keyboard, write a program in Shell Script to calculate the sum of its digits.

6. Write a program in Shell Script to find the factorial of any number entered through the keyboard.

7. Write a program in Shell Script to generate Fibonacci series up to N^{th} term.

8. Write a program in Shell Script to check whether a number is Armstrong number or not.

9. Write a program in Shell Script to check whether a number is Palindrome number or not.

10. Write a program in Shell Script to check whether a number is Prime number or not.

11. Write a program in Shell Script to store integer elements into two “two–dimensional” array and add the two arrays.

12. Write a program in Shell Script to store integer elements into two “two–dimensional” array and multiply the two arrays.

13. Write a program in Shell Script to find average waiting time and average turnaround time of ‘N’ number of processes using First–Come, First–Served (FCFS) Scheduling algorithm. The program should input total number of processes and their corresponding burst time from user.

14. Write a program in Shell Script to find average waiting time and average turnaround time of ‘N’ number of processes using Shortest–Job–First (SJF) Scheduling algorithm. The program should input total number of processes and their corresponding burst time from user.

15. Write a program in Shell Script to find average waiting time and average turnaround time of ‘N’ number of processes using Priority (P) Scheduling algorithm. The program should input total number of processes and their corresponding burst time from user.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

MJ–10 (Th): Digital Logic Design

4 Credits | 60 Minimum Class Hours | Semester V

Objective:

The objective of the course is to enable students to–

- Understand how to represent Binary, Octal, Decimal, and Hexadecimal data and perform the conversion among different number systems.
- Understand the application of Logic Circuit and Boolean algebra in Computer Science and Applications.
- Understand the design of various functional units and digital components of a computer.

Learning Outcome:

After completion of this course, a student will be able to–

- Analyze the performance of commercially available computers.
- Build simple logic circuits using basic logic gates such as AND, OR, NOT, NAND, and NOR.
- Understand the architecture of various digital components, like Integrated Circuits, Decoders, Encoders, Multiplexers, De–multiplexers, Registers, Shift Registers, and Binary Counters.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Mark		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=75

Semester Examination and Distribution of Marks

End Semester Examination (ESE) : 75 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -‘B’			
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

Semester Internal Examination (SIE) : 25 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
Group -‘B’			
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

Unit	Topic	Minimum Class Hours
I	Data Representation	10
II	Digital Logic Circuits	10
III	Combinational Circuits and Flip–Flops	15
IV	Digital Components	25
	Total	**60**

Detailed Syllabus

Unit I: Data Representation (10 Hours)

Number System**:** Binary number system, Octal number system, Decimal number system, Hexadecimal number system, Conversion from one number system to another, Binary Arithmetic (Addition, Subtraction, Multiplication, and Division).

Fixed–Point Number (i.e., Integer) Representation**:** Unsigned integers, Signed integers (Sign–magnitude, 1's complement, and 2's complement representation).

Floating–Point Number Representation**:**32–bit single–precision floating–point numbers, 64–bit double–precision floating–point numbers.

Character Encoding**:** Bit, Byte, Word, BCD, EBCDIC, ASCII, ANSI, Unicode, UTF, ISCII.

Unit II: Digital Logic Circuits (10 Hours)

Logic Circuit**:** Logic Gates (AND, OR, NOT, NAND, NOR, Exclusive–OR, Exclusive–NOR), Converting expressions to logic circuits.

Boolean Algebra**:** Fundamental concepts of Boolean algebra, Postulates of Boolean algebra, Representation of Boolean expressions using truth tables, The principle of Duality/Perfect induction, De-Morgan's theorem, Simplification of Boolean expression, Canonical forms for Boolean expressions (Sum–of–Product and Product–of–Sum), Conversion between canonical forms.

Unit III: Combinational Circuits and Flip-Flops (15 Hours)

Combinational Circuits**:** Half adder, Full adder, Parallel adder and Serial adder.

Flip–Flops**:** Latches, Edge triggered flip–flops (SR flip–flops, D flip–flops, JKflip–flops, and T flip–flops), Pulse triggered flip–flops (Master slave JK flip–flop), Timing diagrams.

Unit IV: Digital Components (25 Hours)

Integrated Circuits (Types of Integrated Circuits Based on Number of Gates, Types of Integrated Circuits Based on Circuit Technology), Decoders, Encoders, Multiplexers, De–multiplexers, Registers {Modes of Operation (SISO, SIPO, PISO and PIPO)}, Shift Registers, Binary Counters {Asynchronous counters (Four bit ripple counter and Decade counter), Synchronous counter (Four bit synchronous counter and Decade counter)}.

Recommended Books:

- M. Morris Mano; **Computer System Architecture** (Third Edition); New Delhi: Prentice–Hall India, 2002
- Donald P Leach, Albert Paul Malvino, GoutamSaha;**Digital Principles and Applications** (Seventh Edition);New Delhi: Tata McGraw HillEducation Pvt. Ltd., 2011
- MostafaAbd–El–Barr, Hesham El–Rewini;**Fundamentals of Computer Organization and Architecture**;John Willy and Sons, Inc. Publication, 2005

- Thomas L. Floyd; **Digital Fundamentals** (Fifth Edition); New Delhi: Pearson Education, 2002

Further readings:

- William Stallings; **Computer Organization and Architecture** (Sixth Edition); New Delhi: Prentice-Hall India, 2002
- B. Ram, Sanjay Kumar; **Computer Fundamentals: Architecture and Organization** (Fifth Edition); New Age International Pvt. Ltd. 2018

◆

MJ–11 (Th): Web Technologies

3 Credits | 45 Minimum Class Hours | Semester V

Objective:

The objective of the course is to enable students to–

- Familiarize the students with markup languages, Cascading Style Sheet, and JavaScript.
- Understand static and dynamic web page development to make the student competent to design static and dynamic websites.

Learning Outcome:

After completion of this course, a student will be able to–

- Design and implement static and dynamic websites with good aesthetic sense of designing using HTML, CSS, and JavaScript.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination														
● Experiments – 15 Marks					● Viva-Voce – 05 Marks					● Practical File – 05 Marks				

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -'B'			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions	1 × 5	5
Group -'B'			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Introduction to Markup Languages	20
II	Cascading Style Sheet	10
III	JavaScript	15
	Total	**45**

Detailed Syllabus

Unit I: Introduction to Markup Languages (20 Hours)

Definition of GML, SGML, VRML, UML, XML, and XHTML.

***HTML Basic*:** Introduction to HTML, The Structure of HTML Document, Global Attributes, Document Structure Tags, Unclosed Tags, Text Formatting Tags, Link Tags, List Tags, Image and Object Tags, Executable Content Tags, Some More Tags (DIV, MARQUEE, NOBR, DFN, HR, COMMENT), HTML Fonts, HTML Color, HTML Elements (HTML Tag vs. HTML Element, Nested HTML Elements), HTML Attributes (Core Attributes, Internationalization Attributes).

***Tables*:** Introduction to HTML Tables, Table Tags, Alignment, Aligning Entire Table, Alignment within a Row, Alignment within a Cell, Attributes, Content Summary, Background Color, Adding a Caption, Setting the Width, Adding a Border, Spacing Within a Cells, Spanning Multiple Rows or Columns, Table Sections and Column Properties, Table as a Design Tool.

***Frames*:** Introduction to Frames and Its Applications, Frames Document, The FRAMESET Tag, Nesting FRAMESET Tag, Placing Content in Frames With The FRAME Tag, Targeting Named Frames, Creating Floating Frames, Using Hidden Frames.

***Forms*:** Creating Forms, The FORM Tag, Named Input Fields, The INPUT Tag, Multiple Lines Text Windows, Drop Down and List Boxes, Hidden Text, Password, File Upload, Button, Submit, Reset, Radio, Checkbox, Select, Option, Forms and Scripting, Action Buttons, Labeling Input Files, Grouping Related Fields, Disabled and Read–only Fields, Form Field Event Handlers, Passing Form Data.

Unit II: Cascading Style Sheet (10 Hours)

Introduction to CSS, CSS Syntax, CSS Selector, How to Add CSS (Inline, Internal, External), CSS Comments, CSS Properties (Background, Border, Display, Float, Font, Line Height, Margin, Opacity, Overflow, Padding, Position, Vertical Align, White Space, Width, Word Wrap, Outline, Visibility, Counter).

Unit III: JavaScript (15 Hours)

Introduction: Introduction to JavaScript, Features of JavaScript, Applications of JavaScript, How to Add JavaScript (Embedding Code, Inline Code, External File).

JavaScript Basics: Comment, Variables (Local Variable, Global Variable), Data Types, Interaction (alert, prompt, confirm), Type Conversions (String Conversion, Numeric Conversion, Boolean Conversion), Operators (Unary minus, Increment, Decrement, Arithmetic Operators, Relational Operators, Logical Operators, Bitwise Operators, Assignment Operators, Special Operators), Selection Structures (‘if’ Statement, ‘if–else’ Statement, ‘if-else if-else’ Statement, ‘switch–case’ statement), Loop Structures (‘while’ loop, ‘do–while’ loop, ‘for’ loop, ‘for–in’ loop), Functions, Function expressions.

JavaScript Objects: Object, Array, String, Date, Math, Number, Boolean, Handler, JSON, RegExp.

UI Events: Mouse events (mousedown/mouseup, mouseover/mouseout, mousemove, click, dblclick), Moving the mouse (mouseover/out, mouseenter/leave), Drag ‘n’ Drop with mouse events, Keyboard (keydown and keyup), Scrolling.

Recommended Books:

- Harvey M. Deitel, Paul J. Deitel, Abbey Deitel; **Internet and World Wide Web: How to Program** (Fifth Edition); Pearson Education, 2012
- Thomas A. Powell, **HTML & CSS: The Complete Reference** (Fifth Edition); McGraw Hill, 2010
- Don Gosselin; **Comprehensive Java Script**; Web Warrior Series, Course Technologies Inc

Further readings:

- Thomas A. Powell; **Web Design: The Complete Reference** (Second Edition); Tata McGraw Hill, 2002
- David Flanagan; **JavaScript The Definitive Guide**; O'Reilly & Associates 1997

* * *

MJ–11 (Pr): Web Technologies Lab

1 Credit | 30 Minimum Class Hours | Semester V

Objective:

The objective of the course is to enable students to–

- Familiarize the students with markup languages, Cascading Style Sheet, and JavaScript.
- Understand static and dynamic web page development to make the student competent to design static and dynamic websites.

Learning Outcome:

After completion of this course, a student will be able to–

- Design and implement static and dynamic websites with good aesthetic sense of designing using HTML, CSS, and JavaScript.

Experiment List

Unit I: Introduction to Markup Languages

1. Create a static website for your family. The home page should contain a suitable name of your family, introduction, postal address, common mobile/land–line number and the picture of the family members. Different heading tags and paragraphs in different face, size, and color of fonts should be applied wherever necessary. Apply all necessary attributes of image tag to the picture of the family members. The names of each family member should be added at the bottom of corresponding picture of the family members, where each of the names should link to a personal page of the corresponding person, i.e., there should be separate personal page about each of the family members including personal information such as name, personal mobile number, e–mail id, date of birth, hobbies and favorite pastimes, etc. Also use the picture of the family members as image maps, where each of the pictures should link to the same personal page of the corresponding person.

2. Create a HTML document to display an ordered list of your top five favorite movies. Each element of the list must have a nested unordered list of songs with name of vocal artist. You may apply text formatting tags to the lists.

3. Design an image map of Indian flag with clickable areas of tricolors. Display respective color information in new HTML page when it is clicked.

4. Create a HTML document to display the Time–Table of your class. Use HTML table tags and the attributes (such as align, border, bgcolor, colspan, rowspan, cellspacing, cellpadding, background, width, and height) wherever necessary. A specimen of the Time–Table is given bellow–

TIME–TABLE

Day/Period	**I** **09:30-10:20**	**II** **10:20-11:10**	**III** **11:10-12:00**	**12:00-12:40**	**IV** **12:40-01:30**	**V** **01:30-02:20**	**VI** **02:20-03:10**	**VII** **03:10-04:00**
Monday	Eng	Mat	Che	**LUNCH**	Lab			Phy
Tuesday	Lab				Eng	Che	Mat	Sports
Wednesday	Mat	Phy	Eng		Che	Library		
Thursday	Phy	Eng	Che		Lab			Mat
Friday	Lab				Mat	Che	Eng	Phy
Saturday	Eng	Che	Mat		Seminar			Sports

5. Create a website that displays the introduction to four courses, viz., Data Mining, Big Data Analytics, Artificial Intelligence, and Information Security in left panel. Add the name of institution with logo in header section, notice board in right panel, and date of creation with developer's name in footer section. After clicking link given in left panel, the appropriate content of that course must be displayed in place of default section. (Use FRAMESET and FRAME Tag). A specimen of the website is given bellow–

<table>
<tr><td colspan="3">Name of Institution with Logo</td></tr>
<tr><td><b><u>Data Mining</u></b>
<b><u>Big Data Analytics</u></b>
<b><u>Artificial Intelligence</u></b>
<b><u>Information Security</u></b></td><td><b>Default Section</b>
After clicking link given in left panel, the appropriate content of that course must be displayed in this section</td><td>News/Other Instructions in MARQUEE Effect</td></tr>
<tr><td>Date of Creation: <Current Date></td><td colspan="2">Developed By <Your Name></td></tr>
</table>

6. Create a HTML page that displays a form with the information – First Name, Last Name, Gender, Subject of Interest, Language Known, Date of Birth (DD–MM–YYYY), Contact No., Address, Pin Code, State, and Country with Reset and Submit button in tabular format.

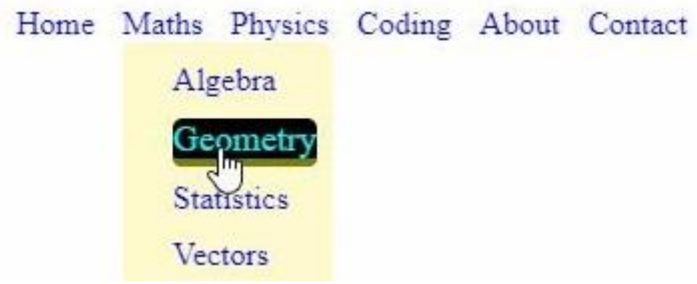
Unit II: Web Page Designing With CSS

1. Create an external CSS named "MyForm.css" with the following style formats–

   - For div tag – border-radius: 5px; background-color: #f2f2f2; padding: 20px;
   - For h1 header – font face: Bookman Old Style, Size: 24pts, color: green;
   - For input tag – width: 100%; padding: 12px 20px; margin: 8px 0; display: inline-block; border: 1px solid #ccc; border-radius: 4px; box-sizing: border-box;
   - For Reset and Register button – width: 100%; background-color: #4CAF50; color: white; padding: 14px 20px; margin: 8px 0; border: none; border-radius: 4px; cursor: pointer;

   Design a web page that contains a div tag. Inside this div tag, place a registration form with the information – Name, Age, Username, Password, and Re–enter Password with Reset and Register button in tabular format to demonstrate the styles defined in the file "MyForm.css".

2. Design a web page that contains a drop–down menu using CSS as follows–



Unit III: JavaScript

1. Design a HTML page for adding two numbers using JavaScript prompt.

2. Create a HTML page that accepts an exponent ‘n’ and a base ‘b’ from the user. Write a JavaScript function to compute the value of b^n and display the result in the same HTML page.

3. Create a HTML page that contains a simple calculator, which can perform the basic arithmetic operations. The calculator will be created using JavaScript.

4. Create a HTML page with rollover menus. The rollover menus must be created using JavaScript.

5. Validate the registration form given in **Unit II** of **Question Number 1** with the following criteria–
   - Name and Age should be Mandatory Fields.
   - Password and Re–enter Password fields should contain same value.
   - Name field should accept only character values.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

IAP: Internship/Apprenticeship/Project

4 Credits | 120 Minimum Class Hours | Semester V

Course Description:

This IAP course aims at providing students with the opportunity to procure practical experience in a professional setting related to their field of study. The students will go in for various tasks such as research, writing, project management, and more, under the guidance of experienced professionals. The program focuses on bridging the gap between academic learning and real-world application, helping students to explore potential career paths. The program will help students in developing essential skills for their careers while also providing them opportunities to actively engage in an on-site experiential learning.

In this Internship/Apprenticeship/Project course the students will have the option to undergo an Apprenticeship or Internship training of 8–week or 2–month duration. The students who do not opt either for Internship or Apprenticeship training will be required to carry out a Field Project based on a topic to be assigned by the concerned department.

Learning Objectives:

- Apply academic knowledge and skills to the real-world tasks and projects in a professional environment.
- Enhance professional skills in research, writing, communication, and project management.
- Gain insight into potential career paths and industries relevant to the student's field of study.
- Strengthen critical thinking and problem-solving skills through engagement with practical challenges and assignments.

Learning Outcomes:

At the end of the Internship/Apprenticeship/Project course, students will be able to–

- Demonstrate the ability to apply academic theories and skills to practical tasks and projects.
- Exhibit proficiency in professional communication, including writing reports, emails, and presenting ideas effectively.
- Manage and complete projects efficiently, demonstrating strong organizational and time–management skills.
- Conduct thorough and efficient research using appropriate methods and sources.
- Build and maintain a network of professional contacts that can support career development.

The Internship/Apprenticeship avenues may include the following:

- Local industry, business organisations, health areas and research laboratories.
- Local governments such as Panchayats, Municipalities and other such bodies, offices of Parliamentarians or other elected representatives, government and non-government social service organisations.
- Media organisations, publication houses, academic institutions, literary and cultural organisations, artists and craft persons.

The students should undertake their Apprenticeship or Internship training preferably in their related discipline in order to enhance their learning.

On successful completion of Internship/Apprenticeship the students will be required to:

- Produce a certificate of having undergone an Internship/Apprenticeship training of 8–week or two–month duration.

- Submit a report of about 40 to 50 pages based on their Internship/Apprenticeship training.

Alternatively, the students who do not opt either for Internship or Apprenticeship training will be required to carry out a Field Project on a topic to be assigned by the concerned department.

The students will prepare their Project/Dissertation of about 40 to 50 pages on their assigned topics under the supervision of a faculty.

Examination and Evaluation: The students opting for Internship/Apprenticeship will prepare a Report and submit the same to the department along with a proper certificate of 8–weeks or 2–month duration.

The students opting to undergo a project or dissertation on the topic assigned by the concerned department will submit a Dissertation/Project report of about 40–50 pages.

The Project Reports/Dissertations submitted by the students will be examined by an external examiner who will also conduct a viva voce examination of the students to assess their understanding of the Dissertation/ Project Report prepared by them.

Internship/Project Report or Project/Dissertation	**75 marks**
Viva-voce	**25 marks**
Total	**100 Marks**

❧

Semester – VI

Programme Structure for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
VI	MN–2C	Minor From Vocational Studies/Discipline – 2 *(To be selected by the students from the list of available options)*		4	
	MJ–12	(Th): Python Programming Language	3–0–1	3	45
		(Pr): Python Programming Language Lab		1	30
	MJ–13	(Th): Data Communication and Computer Network-I	3–1–0	4	60
	MJ–14	(Th): Computer Organization and Architecture	3–1–0	4	60
	MJ–15	(Th): Web Development using JSP	3–0–1	3	45
		(Pr): JSP Lab		1	30
			Total Credits	**20**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **MN–2** (Minor From Discipline–1, **MN–2** (Minor From Vocational Studies/Discipline–2), **MJ** (Major Disciplinary/Interdisciplinary Courses).

MJ–12 (Th): Python Programming Language

3 Credits | 45 Minimum Class Hours | Semester VI

Objectives:

The objective of this course is to introduce the concepts of python programming. This course will help students to learn the python programming from basic to advanced level. The main objectives of the course are as follows–

- To Introduce Python Programming Language as Multipurpose Programming Language with Features and Applications.
- To Practice Basic Language Features of Python.
- Learn core python structures and flow control, Create and run python functions.
- Explore the python library functions for various purposes.
- Object Oriented Programming using Python.
- Learn Files Handling in Python.

Learning Outcomes:

At the end of the course, students will be able to–

- Understand and explain various features of Python language.
- Express different Decision Making statements and Functions.
- Design and Develop Python applications for data analysis using object-oriented concept.
- Understand and summarize different File handling operations.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination ● Experiments – 15 Marks ● Viva-Voce – 05 Marks ● Practical File – 05 Marks														

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -'B'			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions	1 × 5	5
Group -'B'			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Introduction and Python Fundamentals	12
II	Sequences-String, List, Tuples and Dictionaries	09
III	Functions, Python Libraries and File Handling	10
IV	Object Oriented Programming, Database Connectivity	14
	Total	**45**

Detailed Syllabus

Unit I: Introduction and Python Fundamentals (12 Hours)

Introduction to Python Programming: History, Features, Application, Working in Default Python Distribution-Interactive and Script Mode (Python IDLE), Running Python program, Python Character Set, Tokens-Keywords, Identifiers(Names), Literals/Values, Operators, Punctuators, Python Program Structure, Comments in Python, Blocks and Indentation.

Variables and Assignments: Creating Variables, Multiple Assignments, Variable Definition, Dynamic Typing. Input and Output in Python – input() and print() functions.

Data Types: Numbers (Integers, Booleans, Floating-Point, Complex), Strings, Lists and Tuples, Sets, Dictionary, Mutable and Immutable types.

Operators & Expressions: Types of Operators-Arithmetic, Assignment, Relational, Identity, Logical, Membership and Bitwise Operators; Evaluating Expressions, Type Casting.

Introduction to Python Standard Library Modules: Importing and using math module, random module, and statistics module.

Flow of Control: Types of Statement, Statement Flow Control, Selection Statements – if, if – else, if – elif, Nested if Statement; Iteration/Looping Statements - for and while Loop, Loop else Statement, Jump Statements (break and continue), Nested Loops, range() and len() functions.

Unit II: Sequences-String, List, Tuples and Dictionaries (09 Hours)

String: Creating and Storing Strings, Basic String operations, Accessing Characters in String by Index Number, String Slicing and Joining, String methods.

Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, The del Statement, Nested lists

Tuples: Creating Tuples, Basic Tuple Operations, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Tuple Methods, Relations between Tuples and Lists.

Dictionary: Creating Dictionary, Accessing and modifying key: value pairs in Dictionaries, Built-In Functions Used on Dictionaries, Dictionary methods, The del Statement.

Unit III: Functions, Python Libraries and File Handling (10 Hours)

Functions: Calling/ Invoking/ Using a Function, Types of Functions, Defining Functions, Arguments and Parameters, Passing Parameters, Returning Values from Functions, Scope of Variables, Mutable/ Immutable Properties of Passed Data Objects, Recursion.

Python Libraries: Modules, Packages, Libraries, Python Standard Library, Importing Modules, Installing Packages with *pip* installer, Creating and Using Libraries/ Packages, Using NumPy, SciPy and Matplotlib library.

File Handling: Types of files-Text files, Binary files and CSV (Comma separated values) files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, The Pickle module, Reading and Writing CSV files.

Unit IV: Object Oriented Programming, Database Connectivity (14 Hours)

Classes and Objects: Introduction, Class Definition, Creating Objects, Instances as Arguments, Instances as return values, Built-in Class Attributes, Constructor, Inheritance, Polymorphism - Method Overloading & Overriding.

Python and MySQL: Installing MySQL Connector, Verifying the Connector Installation, Using MySQL from Python, Retrieving all rows from a table, Inserting rows into a table, Deleting rows from table, Updating rows in a table, Creating database tables through Python.

Recommended Books:

- E. Balagurusamy; **Introduction to Problem Solving with Python**; TMH
- ReemaThareja; **Problem Solving and Programming with Python**; Oxford University Press

Further Readings:

- JohnV. Guttag; **Introduction to Computation and Programming using Python**; PHI Learning
- Jason Montojo, Jennifer Campbell, Paul Gries; **An Introduction to Computer Science using Python 3**; SPD

Web Resources:

- https://www.learnpython.org/
- https://nptel.ac.in/courses/106/106/106106212/
- Python Tutorial – https://docs.python.org/3/tutorial/index.htm

* * *

MJ–12 (Pr): Python Programming Language Lab

1 Credit | 30 Minimum Class Hours | Semester VI

Objective:

The objective of the course is to enable students to–

- To implement the python programming features in practical applications.
- To write, test, and debug simple Python programs.
- To implement Python programs with conditionals and loops.
- Use functions for structuring Python programs.
- Represent compound data using Python lists, tuples, dictionaries and modules.

Learning Outcome:

After completion of this course, a student will be able to–

- Understand the numeric or real life application problems and solve them.
- Use Python lists, tuples, dictionaries for representing compound data.
- Develop Python programs step-wise by defining functions and calling them.
- Apply the best features available in Python to solve the situational problems.

Experiment List

Unit I: Introduction and Python Fundamentals	
1.	Write a program to demonstrate different datatypes in python. Show data type and id of various data elements.
2.	Write a program to perform different arithmetic operations on numbers in python.
3.	Write a program to swap two numbers without any arithmetic operator or temporary variable.
4.	Write a python program to input three numbers and display the sum and difference of largest and smallest number.
5.	Write a python script to display first N Fibonacci series numbers starting from 0 and 1. Print ***'p'*** as suffix to those numbers which are also prime. For example – *For N=10* *Fibonacci series : 0,1,1,2p,3p,5p,8,13p ,21,34*
6.	Write a program to find sum of the series without using math module or exponent (**) operator: $1+x-x^2/2! + x^3/3! – x^4/4! + \ldots\ldots\ldots\ldots x^n/n!$
Unit II: Sequences-String, List, Tuples and Dictionaries	
7.	Write a python program to check whether the given string is palindrome or not.
8.	Create a list and perform the following methods (a) insert() (b) remove() (c) append() (d) len() (e) pop() (f)clear()

9.	Write a program to input n numbers into a tuple and calculate mean, median and mode of tuple's element.
10.	Write a program that repeatedly asks the user to enter product names and prices. Store all of these in a dictionary whose keys are the product name and whose values are the prices. When the user is done entering products and prices, allow them to repeatedly enter a product name and print the corresponding price or a message if the product is not in the dictionary.
Unit III: Functions, Python Libraries and File Handling	
11.	Write a function that takes a number ***n*** and then returns a randomly generated number having exactly ***n*** digits e.g., if ***n*** is 2 then function can randomly return a number between 10 and 99.
12.	Write a function ***convertNum (dec, base)*** that receives a decimal number and base and prints the equivalent number in other number bases i.e., binary, decimal and hexadecimal. The program should check that base is 2, 8 or 16, otherwise display the message "Invalid base".
13.	Write a python program to find factorial of a number using recursion.
14.	Write a python program to create a package (***MyMath***), having modules ***Basic*** and ***Advance***. The ***Basic*** module contains functions sum(a,b) and avg(a,b) to return sum and average of two numbers respectively and ***Advance*** module contains the function square(a) and cube(a) to return square and cube respectively. Import the modules from ***MyMath*** package into a program and perform basic operations.
15.	Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
16.	Write a program to create a binary file "Book.dat" having structure [BookNo, Book_Name, Author, Price]. • Write a user defined function CreateFile() to input data for a record and add to Book.dat . • Write a function ShowBooks() to display details of all books. • Write a finctionSearchBook(BookNo) to search a book. • Write a function ModifyBooks(BookNo, Price) which accept bookno and price in argument and increment the price of that book
17.	Write a program to create a CSV file "user.csv" which will contain user name and passwordfor some entries. Read the "user.csv" file and display all records.
Unit IV: Object Oriented Programming, Database Connectivity	
18.	Create a Student class and initialize it with ***name***, ***roll,*** and ***age***. Make methods to : (a) Display - It should display all information of the student. (b) setAge - It should assign age to student (c)setMarks - It should assign marks to the student.
19.	Write a python program to create ***Account*** class with ***deposit***, ***withdraw*** function and then implement it. Use constructor to initialize ***initial balance***. Inherit a class ***Customer*** from ***Account*** having fields account number and customer name. Override a method *display()* of Account class into Customer to display customer details with balance amount.

20.	Write a complete application on database operations on student data such as [Roll, Name, Marks]. The program should perform following task – (a) Connection to the database (b) Creating the student table (c) Saving / Inserting records (d) Reading records (e) Updating records (f) Deleting records

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

MJ–13 (Th): Data Communication and Computer Network – I

4 Credits | 60 Minimum Class Hours | Semester VI

Objectives:

The objective of the course is to introduce the concepts of computer networks, physical, and data link layer. This course will help students to learn the fundamental layered structure of networks, understand common offered layered services, and examine protocols used to operate the network.

Learning Outcome:

After completion of this course, a student will be able to–

- Understand about networks, internal components, and its functionality.
- Gain basic knowledge about the technologies/protocols involved in the physical and data link layer.

Outline of the Course

Minimum Class Hours		**Exam Time (Hours)**		**Credits**		**Marks**									
						Semester Internal		**End Semester**		**Full Mark**		**Pass Marks**		**Total Marks**
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=100

Semester Examination and Distribution of Marks

<u>End Semester Examination (ESE) : 75 Marks</u>

S.No	**Questions**		**Marks**
	Group -‘A’		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -‘B’		
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

<u>Semester Internal Examination (SIE) : 25 Marks</u>

S.No	**Questions**		**Marks**
	Group -‘A’		
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
	Group -‘B’		
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

Unit	**Topic**	**Minimum Class Hours**
I	Overview	20
II	Physical Layer and Media	20
III	Data Link Layer	20
	Total	**60**

Detailed Syllabus

Unit I: Overview (20 Hours)

Data Communication, Components of Data Communication, Direction of Data Flow (Simplex Mode, Half–duplex Mode, Full–duplex Mode), Communication Task, Computer Networks, Network Criteria, Physical Structures [Type of Connection (Point–to–Point, Multipoint), Network Topology {Physical Topology, Logical Topology (Bus, Ring, Star, Mesh, Tree, Hybrid)}], Categories of Networks (Personal Area Network, Local Area Network, Wireless Local Area Network, Metropolitan Area Network, Wide Area Network, Storage Area Network, System Area Network, Passive Optical Local Area Network, Enterprise Private Network, Virtual Private Network), The Internet (A Brief History, Today's Internet), Computer Network Protocol, The OSI Model, TCP/IP Protocol Suite.

Unit II: Physical Layer and Media (20 Hours)

***Data and Signals*:** Analog and Digital (Analog and Digital Data, Analog and Digital Signals), Periodic Analog Signals (Wavelength, Bandwidth), Digital Signals (Bit Rate, Bit Length), Transmission Impairments (Attenuation, Distortion, Noise), Performance (Bandwidth, Throughput, Latency, Bandwidth–Delay Product, Jitter).

***Digital Transmission and Multiplexing*:** Transmission Modes (Parallel Transmission, Serial Transmission), Multiplexing (Frequency–Division Multiplexing, Wavelength–Division Multiplexing, Synchronous Time–Division Multiplexing, Statistical Time–Division Multiplexing).

***Transmission Media*:**Guided Media {Twisted–Pair Cable (Unshielded Twisted–Pair, Shielded Twisted–Pair), Co–Axial Cable, Fiber–optic Cable (Single Propagation Mode, Multiple Propagation mode)}, Unguided Media (Radio Wave Communication, Micro Wave Communication, Infrared Wave Communication).

Unit III: Data Link Layer (20 Hours)

***Error Detection and Correction*:** Introduction (Types of Errors, Redundancy, Detection versus Correction, Forward Error Correction versus Retransmission, Coding).

Data Link Control: Framing (Fixed–Size Framing, Variable–Size Framing), Flow Control, Error Control, Protocols {Noiseless Channels (Simplest Protocol, Stop–and–Wait Protocol), Noisy Channels (Stop–and–Wait Automatic Repeat Request, Go–Back–NAutomatic Repeat Request, Selective Repeat Automatic Repeat Request)}.

Wireless LAN, Connecting LAN: Bluetooth Technology, Connecting Devices (Passive Hubs, Repeaters, Active Hubs, Bridges, Two–Layer Switches, Routers, Three–Layer Switches, Gateway).

Wireless WAN: Cellular Telephony (Frequency–Reuse Principle, Transmitting, Receiving, Roaming, First Generation, Second Generation, Third Generation), Satellite Networks {Orbits, Footprint, Three Categories of Satellites (GEO, MEO, LEO)}.

Recommended Books:

- Behrouz A. Forouzan; **Data Communications and Networking** (Fourth Edition); McGraw–Hill, 2007
- Andrew S. Tanenbaum; **Computer Networks** (Fourth Edition); Prentice Hall, 2003
- William Stallings; **Data & Communications** (Sixth Edition); Pearson Education

Further readings:

- Prakash C. Gupta; **Data Communications & Computer Networks**; PHI, New Delhi

- U. D. Black; **Data Communications and Distributed Networks** (Third Edition); Prentice Hall of India, 1993
- Fred Halsall; **Data Communication, Computer Networks and Opens Systems** (Fourth Edition); Pearson Education, 2000

◆

MJ–14 (Th): Computer Organization and Architecture

4 Credits | 60 Minimum Class Hours | Semester VI

Objective:

The objective of the course is to introduce some of the basic understanding of register transfer and microoperations, basic computer organization and design of a digital computer. The aim is to give an overview of central processing unit of the computer and its functions.

Learning Outcome:

After completion of this course, a student will be able to–

- Analyze the performance of commercially available computers.
- Understand register transfer, microoperations, basic computer organization and design of a digital computer.
- Have a thorough understanding of the basic structure and operation of CPU including instruction formats, addressing modes, etc. Students will identify the elements of modern instructions sets and their impact on processor design.
- Have a better understanding and utilization of digital computers.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=100

Semester Examination and Distribution of Marks

End Semester Examination (ESE) : 75 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -‘B’		
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

Semester Internal Examination (SIE) : 25 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
	Group -‘B’		
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

Unit	Topic	Minimum Class Hours
I	Register Transfer and Microoperations	20
II	Basic Computer Organization and Design	15
III	Central Processing Unit	25
	Total	**60**

Detailed Syllabus

Unit I: Register Transfer and Microoperations (20 Hours)

***Register Transfer*:** Definition of Register Transfer Language, Register Transfer, Bus Transfer, Memory Transfer.

***Microoperations*:** Arithmetic Microoperations (Binary Adder, Binary Adder–Subtractor, Binary Increment, Arithmetic Circuit), Logic Microoperations (List of logic operation), Arithmetic Logic Shift Unit.

Unit II: Basic Computer Organization and Design (15 Hours)

General System Architecture (Store Program Control Concept and Flynn's Classification of Computers), Issues in Computer Design, Computer Architecture versus Computer Organization, Computer Registers, Computer Instructions, Instruction Cycle, Instruction Types (Data Transfer, Arithmetic, Logical, String Manipulation, Control Transfer, Loop Control, and Processor Control).

Unit III: Central Processing Unit (25 Hours)

An Overview of Central Processing Unit (CPU) with Block Diagram, General Register Organization, Control Word, Stack Organization (Register Stack and Memory Stack), Arithmetic Logic Unit (ALU), Data Path, Design of Control Unit (Hardwired Control and Microprogrammed Control), Instruction Formats (Zero, One, Two and Three Address Instruction), Addressing Modes, Program Interrupt, Types of Interrupts, Characteristics of Complex Instruction Set Computer (CISC) and Reduced Instruction Set Computer (RISC).

Recommended Books:

- M. Morris Mano; **Computer System Architecture** (Third Edition); New Delhi: Prentice–Hall India, 2002
- Donald P Leach, Albert Paul Malvino, Goutam Saha; **Digital Principles and Applications** (Seventh Edition); New Delhi: Tata McGraw Hill Education Pvt. Ltd., 2011
- Mostafa Abd–El–Barr, Hesham El–Rewini; **Fundamentals of Computer Organization and Architecture**; John Willy and Sons, Inc. Publication, 2005
- Thomas L. Floyd; **Digital Fundamentals** (Fifth Edition); New Delhi: Pearson Education, 2002
- M. Morris Mano; **Digital Logic and Computer Design**; New Delhi: Prentice-Hall India, 1994

Further readings:

- William Stallings; **Computer Organization and Architecture** (Sixth Edition); New Delhi: Prentice-Hall India, 2002
- B. Ram, Sanjay Kumar; **Computer Fundamentals: Architecture and Organization** (Fifth Edition); New Age International Pvt. Ltd. 2018
- V. Carl Hamacher, Zvonko G. Vranesic, Saftwat G. Zaky; **Computer Organization** (Fourth Edition); New Delhi: Tata McGraw–Hill, 1996

* * *

MJ–15 (Th): Web Development using JSP

3 Credits | 45 Minimum Class Hours | Semester VI

Objective:

The objective of the course is to enable students to–

- Understand the concepts underlying technologies in JSP.
- Install and configure Java Development Kit (JDK), Web Server (Apache Tomcat Server), and Database (MySQL or, Oracle).
- Web application development is also being discussed to make the student competent to design dynamic and secure websites.

Learning Outcome:

After completion of this course, a student will be able to–

- Develop web application that has direct relevance to the real world using JSP, MySQL, Oracle, and Apache Tomcat Server in MVC architecture.
- Design and implement database connectivity using servlet programming.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks									
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination ● Experiments – 15 Marks ● Viva-Voce – 05 Marks ● Practical File – 05 Marks														

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -'B'		
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions	1 × 5	5
	Group -'B'		
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Overview of JSP	5
II	Elements, Directives, Implicit Objects, and Standard Action Tags	10
III	Expression Language (EL) and Exception Handling	15
IV	Advanced JSP	15
	Total	**45**

Detailed Syllabus

Unit I: Overview of JSP (5 Hours)

Brief Introduction to J2EE Architecture, Introduction to JSP, Architecture of JSP, Lifecycle of JSP, Advantages of JSP over Servlet, Applications of JSP, Installation and Configuration of Java Development Kit (JDK), Installation and Configuration of Apache Tomcat Server, Installation and Configuration of MySQL.

Unit II: Elements, Directives, Implicit Objects, and Standard Action Tags (10 Hours)

Elements: Declaration Tag, Scriptlet Tag, Expression Tag, Comments, Creating a simple JSP Page, How to run a simple JSP Page?, The Directory Structure of JSP.

Directives: Page Directive, Include Directive, Taglib Directive.

Implicit Objects: out, request, response, config, application, session, pageContext, page, exception.

Standard Action Tags: jsp:useBean, jsp:include, jsp:setProperty, jsp:getProperty, jsp:forward, jsp:plugin, jsp:attribute, jsp:body, jsp:text, jsp:param, jsp:output.

Unit III: Expression Language (EL) and Exception Handling (15 Hours)

JSP As An Expression Language: Syntax of Expression Language, Selection Control Structures (‘if’ Statement, ‘if–else’ Statement, ‘if-else if-else’ Statement, ‘switch–case’ statement), Loop Control Structures (‘while’ loop, ‘do–while’ loop, ‘for’ loop, ‘for–in’ loop), Operators {Unary minus, Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Special Operators (., (), [], Empty)}, Implicit Objects in Expression Language (pageScope, requestScope, sessionScope, applicationScope, param, paramValues, header, headerValues, initParam, cookie, and pageContext).

Exception Handling: Checked Exception, Runtime Exception, Error Exception.

Unit IV: Advanced JSP (15 Hours)

Java Database Connectivity: Introduction to Java Database Connectivity (JDBC), JDBC drivers (JDBC–ODBC Bridge Driver, Native Driver, Network Protocol Driver, and Thin Driver), JDBC API (Popular interfaces and classes), Advantages of JDBC API over ODBC API, Steps to Connect Java Application with Database, Java Database Connectivity with MySQL, Java Database Connectivity with Oracle.

Servlet Basics: Introduction to Servlet, Servlet API, Servlet Life Cycle, How Servlet Works?, Servlet Example by Inheriting HttpServlet Class.

Action Elements (HTTP Client Request, HTTP Server Response, HTTP Status Codes), Cookies Handling, Form Processing {GET Method, POST Method, getParameter(), getParameterValues(), getParameterNames(), getInputStream()}, Filter Mapping in web.xml with Servlet.

Recommended Books:

- Phil Hanna; **JSP: The Complete Reference**; New York: Osborne/McGraw-Hill
- Stephen Asbury, Scot R. Weiner, Wiley; **Developing Java Enterprise Applications**; 1998

◆

MJ–15 (Pr): JSP Lab

1 Credit | 30 Minimum Class Hours | Semester VI

Objective:

The objective of the course is to enable students to–

- Understand the concepts underlying technologies in JSP.
- Install and configure Java Development Kit (JDK), Web Server (Apache Tomcat Server), and Database (MySQL or, Oracle).
- Web application development is also being discussed to make the student competent to design dynamic and secure websites.

Learning Outcome:

After completion of this course, a student will be able to–

- Develop web application that has direct relevance to the real world using JSP, MySQL, Oracle, and Apache Tomcat Server in MVC architecture.
- Design and implement database connectivity using servlet programming.

Experiment List

Unit I: Elements, Directives, Implicit Objects, and Standard Action Tags

1. Create a JSP document named “registration.jsp” that displays a form with the information – First Name, Last Name, Gender, Subject of Interest, Language Known, Date of Birth (DD–MM–YYYY), Contact No., Address, Pin Code, State, and Country with Reset and Submit button in tabular format. After clicking on ‘Submit’ button, all the information should display on another JSP document named “success.jsp” with a success message using Implicit Object ‘request’.

2. Create a simple JSP document to redirect the user to the official website of ISRO https://www.isro.gov.in/ using Implicit Object ‘response’.

3. Create a JSP document to find out the following using Implicit Object ‘session’–

   (i) Unique identifier assigned to the session
   (ii) Time of session creation
   (iii) Last time the client sent a request associated with that session
   (iv) If the client is new to that session
   (v) Number of visits by a client to the web page.

Unit II: Expression Language (EL) and Exception Handling

1. Create a JSP document named “calculator.jsp” that displays a form with the following information in tabular format–

   (i) Four radio buttons to perform different mathematical operations such as addition, subtraction, multiplication, and division. By default, the radio button for addition operation will be checked.
   (ii) Two text boxes to take two numbers from user.
   (iii) Two buttons such as ‘Reset’ to reset all the input fields to its initial values and ‘Show Result’ to show the result of operation.

   After clicking on ‘Show Result’ button, the result of mathematical operations should display on another JSP document named “process.jsp” using Expression Language (EL).

2. Create a JSP document named "registration.jsp" that displays a form with the information – First Name, Last Name, Gender, Subject of Interest, Language Known, Date of Birth (DD–MM–YYYY), Contact No., Address, Pin Code, State, and Country with Reset and Submit button in tabular format. After clicking on 'Submit' button, all the information should display on another JSP document named "success.jsp" with a success message using Expression Language (EL).

Unit III: Advanced JSP

1. Create a table 'Student' with field names 'StudentId', 'Name', 'Email', 'ContactNo' in either MySQL or Oracle database. Create a JSP document to fetch all the records from the table 'Employee' and display them.

2. Write a Servlet program to store a cookie in user's device. Cookie name should be 'sessionid' and cookie value should be id of that session. Retrieve the cookie from user's device and display the name and value of it.

3. Create a table 'Professor' with field names 'TeacherId', 'Name', 'Email', 'ContactNo' in either MySQL or Oracle database. Design an HTML form in JSP document to invoke a Servlet which should catch all the fields of table 'Professor' from the client side and display the same to client side in tabular format using deployment descriptor "web.xml" to map the Servlet.

Unit IV: Web Application Development using JSP

1. Develop Java Server Pages (JSP) to create Registration Form, Login Form, File Uploading, File Downloading, Handling Date, Page Redirect, Hits Counter, and Sending Email using MVC Architecture.

2. Students should develop a **Mini Project** that has some direct relevance to the real world, using Java Technology (JSP, Servlets, JDBC) as front end tools and MySQL with Apache Tomcat server as back end tools. Every student should develop the project individually and not in a group.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

Semester – VII

Programme Structure for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
VII	MN–1D	Minor From Discipline–1 *(To be selected by the students from the list of available options)*		4	
	MJ–16	(Th): Data Communication and Computer Network-II	3–1–0	4	60
	MJ–17	(Th): Computer Oriented Numerical Methods	3–1–0	4	60
	MJ–18	(Th): Computer Graphics	3–0–1	3	45
		(Pr): Computer Graphics Lab		1	30
	MJ–19	(Th): Information Security	3–0–1	3	45
		(Pr): Information Security Lab		1	30
			Total Credits	**20**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **AEC** (Ability Enhancement Course), **VAC** (Value Added Course), **SEC** (Skill Enhancement Course), **MDC** (Multi Disciplinary Course), **MN–1** (Minor From Discipline–1, **MJ** (Major Disciplinary/Interdisciplinary Courses).

MJ–16 (Th): Data Communication and Computer Network – II

4 Credits | 60 Minimum Class Hours | Semester VII

Objectives:

This course is a continuation of the previous semester. The objective of the course is to introduce network, transport, and application layer. This course will help students to learn the routing and congestion control algorithms, logical addressing, elements of transport protocols, domain name system, remote logging, electronic mail, and file transfer.

Learning Outcomes:

After completion of this course, a student will be able to understand about the routing and congestion control algorithms, logical addressing, elements of transport protocols, domain name system, remote logging, electronic mail, and file transfer.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=100

Semester Examination and Distribution of Marks

End Semester Examination (ESE) : 75 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -‘B’			
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

Semester Internal Examination (SIE) : 25 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
Group -‘B’			
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

Unit	Topic	Minimum Class Hours
I	Network Layer	25
II	Transport Layer	15
III	Application Layer	20
	Total	**60**

Detailed Syllabus

Unit I: Network Layer (25 Hours)

***Network Layer Design Issues*:** Store–and–Forward Packet Switching, Services Provided to the Transport Layer, Implementation of Connectionless Service, Implementation of Connection–Oriented Service.

***Routing Algorithms*:** Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing, Routing for Mobile Hosts, Routing in Ad Hoc Networks.

Congestion Control Algorithms: General Principles of Congestion Control, Load Shedding, Jitter Control.

Logical Addressing: Internet Protocol (IP) addresses, Subnets, IPv4 Addresses (Address Space, Notations, Classful Addressing, Classless Addressing, Network Address Translation), IPv6 Addresses (Structure, Address Space).

Unit II: Transport Layer (15 Hours)

Services Provided to the Application Layer, Elements of Transport Protocols (Addressing, Connection Establishment, Connection Release, Flow Control and Buffering, Multiplexing, Crash Recovery), Internet Transport Protocols (User Datagram Protocol, Transmission Control Protocol).

Unit III: Application Layer (20 Hours)

***Domain Name System*:** Introduction, Name Space (Flat Name Space, Hierarchical Name Space), Domain Name Space (Label, Domain Name, Domain), Domain Name System in The Internet (Generic Domains, Country Domains, Inverse Domain), Resolution (Resolver, Mapping Names to Addresses, Mapping Address to Names, Recursive Resolution, Iterative Resolution, Caching).

Remote Logging, Electronic Mail, and File Transfer: Remote Logging (TELNET), Electronic Mail (Architecture, User Agent, Message Transfer Agent: SMTP, Message Access Agent: POP and IMAP, Web–Based Mail), File Transfer (File Transfer Protocol).

Recommended Books:

- Behrouz A. Forouzan; **Data Communications and Networking** (Fourth Edition); McGraw–Hill, 2007
- Andrew S. Tanenbaum; **Computer Networks** (Fourth Edition); Prentice Hall, 2003
- William Stallings; **Data & Communications** (Sixth Edition); Pearson Education

Further readings:

- Prakash C. Gupta; **Data Communications & Computer Networks**; PHI, New Delhi
- U. D. Black; **Data Communications and Distributed Networks** (Third Edition); PHI, 1993
- Fred Halsall; **Data Communication, Computer Networks and Opens Systems** (Fourth Edition); Pearson Education, 2000

MJ–17 (Th): Computer Oriented Numerical Methods

4 Credits | 60 Minimum Class Hours | Semester VII

Objective:

The objective of the course is to provide the numerical methods of solving the non–linear equations, interpolation, differentiation, and integration. To improve the student's skills in numerical methods by using the numerical analysis software and computer facilities.

Learning Outcomes:

At the end of the course, students will be able to:

- Apply numerical methods to solve equations, interpolation, differentiation, integration, and differential equations.
- Analyze and minimize computational errors in numerical methods.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=100

Semester Examination and Distribution of Marks

End Semester Examination (ESE) : 75 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -'B'			
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

Semester Internal Examination (SIE) : 25 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
Group -'B'			
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

***Note**: Students will be allowed to use non-programmable scientific calculator in the End Semester External and Internal Examinations. However, sharing of calculator will not be permitted in the examination.*

Unit	Topic	Minimum Class Hours
I	Errors in Numerical Calculations	5
II	Solution of Algebraic and Transcendental Equations	15
III	Interpolation	15
IV	Numerical Differentiation and Integration	15
V	Numerical Solution of Ordinary Differential Equations	10
	Total	**60**

Detailed Syllabus

Unit I: Errors in Numerical Calculations (05 Hours)

Introduction, Numbers and their accuracy, Errors and their Computations- Absolute, Relative and Percentage Error.

Unit II: Solution of Algebraic and Transcendental Equations (15 Hours)

Introduction, Types of Non-Linear Equations: Polynomial equations, Transcendental equations; Methods of Solving Non-Linear Equations: Direct Methods, Iterative Methods (Bisection Method, Newton-Raphson Method, Method of False Position and Secant Method).

Unit III: Interpolation (15 Hours)

Introduction, Finite Differences-Forward, Backward and Central Differences, Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Lagrange's Interpolation Formula.

Unit IV: Numerical Differentiation and Integration (15 Hours)

Introduction, Numerical Differentiation, Numerical Integration – Trapezoidal Rule, Simpson's 1/3 Rule, Simpson's 3/8 Rule.

Unit V: Numerical Solution of Ordinary Differential Equations (10 Hours)

Solution by Taylor's Series, Euler's method

Recommended Books:

- S.S. Sastry; **Introductory methods of Numerical Analysis** (Fifth Edition); Prentice Hall of India, New

Further readings:

- B.S. Grewal; **Numerical Methods in Engineering & Science**; Khanna Publishers, Delhi
- S.C. Gupta and V.K. Kapoor; **Elements of Mathematics, Statistics**; Sultan Chand and Sons
- Jaan Kiusalaas; **Numerical Methods in Engineering with Python**; Cambridge

◆

MJ–18 (Th): Computer Graphics

3 Credits | 45 Minimum Class Hours | Semester VII

Objective:

The objective of the course is to enable students to–

- Understand the theoretical background for introductory computer graphics and demonstrates the application of computer science to graphics.
- Gain knowledge about graphics hardware devices and software used.
- Understand the two–dimensional and three–dimensional graphics and their transformations.
- Become familiar with understand clipping techniques.

Learning Outcome:

After completion of this course, a student will be able to–

- Explain the basic principles, algorithms, and mathematical foundations of computer graphics.
- Understand the architecture and working of graphics hardware and software.
- Implement fundamental algorithms for line drawing, circle drawing, and polygon filling.
- Utilize algorithms for 2D and 3D transformations, including translation, rotation, scaling, and viewing transformations.
- Apply viewing pipeline concepts for 2D and 3D graphics.
- Implement clipping algorithms like Cohen–Sutherland and Sutherland–Hodgman for efficient rendering.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination														
● Experiments – 15 Marks					● Viva-Voce – 05 Marks					● Practical File – 05 Marks				

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -'B'			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions	1 × 5	5
Group -'B'			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Overview of Graphics System	18
II	Two–Dimensional Geometric Transformations and Viewing	15
III	Three–Dimensional Geometric Transformations and Viewing	12
	Total	**45**

Detailed Syllabus

Unit I: Overview of Graphics System (18 Hours)

Introduction: Video Display Devices, Refresh Cathode–Ray Tubes, Raster–Scan Display, Random–Scan Display, Input Devices, Hard–Copy Devices, Graphics Software.

Output Primitives: Points, Line–Drawing Algorithms (DDA and Bresenham's Line Drawing Algorithm), Circle–Generating Algorithms (Bresenham's and Midpoint Circle Algorithms), Ellipse–Generating Algorithms (Midpoint Ellipse Algorithm), Filled–Area Primitives (Scan–Line Polygon Fill Algorithm, Boundary–Fill Algorithm, Flood–Fill Algorithm).

Unit II: Two–Dimensional Geometric Transformations and Viewing (15 Hours)

Two–Dimensional Geometric Transformations: Basic Transformations, Matrix Representations and Homogeneous Coordinates, Composite Transformations, Reflection and Shear, Transformations between Coordinates Systems, Raster Methods for Transformations.

Two–Dimensional Viewing: The Viewing Pipeline, Viewing Coordinate Reference Frame, Window–to–Viewport Coordinate Transformation, Clipping Operations {Point Clipping, Line Clipping (Cohen–Sutherland Line Clipping and Liang–Barsky Line Clipping), Polygon Clipping (Sutherland–Hodgeman Polygon Clipping, Weiler–Atherton Polygon Clipping)}.

Unit III: Three–Dimensional Geometric Transformations and Viewing (12 Hours)

Three–Dimensional Geometric Transformations: Translation, Rotation, Scaling, Reflection and Shears, Composite Transformations, Modeling and Coordinate Transformations.

Three–Dimensional Viewing: Viewing Pipeline, Viewing Coordinates, Projections and Clipping.

Recommended Books:

- Phil Hanna; **JSP: The Complete Reference**; New York: Osborne/McGraw-Hill
- Stephen Asbury, Scot R. Weiner, Wiley; **Developing Java Enterprise Applications**; 1998

◆

MJ–18 (Pr): Computer Graphics Lab

1 Credit | 45 Minimum Class Hours | Semester VII

Objective:

The objective of the course is to enable students to–

- Understand the practical implementation of line, circle, and eclipse drawing algorithms.
- Understand the practical implementation of filled–area primitives.
- Understand the two–dimensional graphics and their transformations.

Learning Outcome:

After completion of this course, a student will be able to–

- Implement line, circle, and eclipse drawing algorithms using C programming language.
- Implement filled–area primitive algorithms using C programming language.
- Perform two–dimensional transformations such as Translation, Scaling, and Rotation using C programming language.
- Implement different types of clipping algorithms using C programming language.

Experiment List

Unit I: Overview of Graphics System

4. Write a program in C programming language to implement DDA (Digital Differential Analyzer) Line drawing algorithm.

5. Write a program in C programming language to implement Bresenham’s Line drawing algorithm.

6. Write a program in C programming language to implement Bresenham’s Circle drawing algorithm.

7. Write a program in C programming language to implement Midpoint Circle drawing algorithm.

8. Write a program in C programming language to implement Midpoint Ellipse drawing algorithm.

9. Write a program in C programming language to implement Scan–Line Polygon Fill algorithm.

10. Write a program in C programming language to implement Boundary–Fill algorithm by 4–connected pixels.

11. Write a program in C programming language to implement Boundary–Fill algorithm by 8–connected pixels.

12. Write a program in C programming language to implement Flood–Fill algorithm to fill polygon by 4–connected pixels.

Unit II: Two–Dimensional Geometric Transformations and Viewing

3. Write a menu–driven program in C programming language to perform Two–Dimensional Transformations such as Translation, Scaling, and Rotation on a triangle.

4. Write a program in C programming language to implement Cohen–Sutherland Line Clipping algorithm.

5. Write a program in C programming language to implement Liang–Barsky Line Clipping algorithm.

6. Write a program in C programming language to implement Sutherland–Hodgeman Polygon Clipping algorithm.

7. Write a program in C programming language to implement Weiler–Atherton Polygon Clipping algorithm.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

MJ–19 (Pr): Information Security

3 Credits | 45 Minimum Class Hours | Semester VII

Objective:

The objective of the course is to enable students to–

- Learn fundamentals of cryptography and its applications.
- Understand the concepts symmetric and asymmetric Cryptographic Techniques.

Learning Outcome:

After completion of this course, a student will be able to–

- Understand and explain the risks faced by computer systems and networks.
- Identify and analyze security problems in computer systems and networks.
- Explain how standard security mechanisms work.
- Develop security mechanisms to protect computer systems and networks.
- Write programs that are more secure.
- Use cryptography algorithms and protocols to achieve computer security.

Outline of the Course

Minimum Class Hours		**Exam Time (Hours)**		**Credits**		**Marks**								
						Semester Internal		**End Semester**		**Full Mark**		**Pass Marks**		**Total Marks**
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100

Marks Distribution of End Semester Practical Examination

● Experiments – 15 Marks ● Viva-Voce – 05 Marks ● Practical File – 05 Marks

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	**Questions**		**Marks**
Group -'A'			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -'B'			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	**Questions**		**Marks**
Group -'A'			
1	Five very short answer type questions	1 × 5	5
Group -'B'			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Introduction to Information Security	06
II	Symmetric Cryptographic Techniques	12
III	Asymmetric Cryptographic Techniques	12
IV	Digital Signatures and Authentication	15
	Total	**45**

Detailed Syllabus

Unit I: Introduction to Information Security (6 Hours)

Introduction, Threat, Attack, Cryptography, Cryptanalysis, The OSI Security Architecture (Security Attack, Security Mechanism, and Security Service).

Unit II: Symmetric Cryptographic Techniques (12 Hours)

***Classical Encryption Techniques*:** Symmetric Cipher Model, Substitution Techniques (Caesar Cipher, Monoalphabetic Ciphers, Playfair Cipher, One-Time Pad), Transposition Techniques.

Stream Ciphers, Block Ciphers, Feistel Cipher, Data Encryption Standard (DES), Advanced Encryption Standard (AES).

Unit III: Asymmetric Cryptographic Techniques (12 Hours)

Public–Key Cryptosystems (Applications and Requirements), Rivest–Shamir–Adleman (RSA) Algorithm.

***Hash Functions*:** Requirements, Simple Hash Functions, Birthday Attacks, Security of Hash Functions (Brute–Force Attacks and Cryptanalysis), Secure Hash Algorithm (SHA).

Unit IV: Digital Signatures and Authentication (15 Hours)

***Digital Signatures*:** Requirements, Direct Digital Signature, Arbitrated Digital Signature.

***Authentication Protocol*:** Mutual Authentication, Symmetric Encryption Approaches, Public–Key Encryption Approaches.

***One–Way Authentication*:** Symmetric Encryption Approach, Public–Key Encryption Approaches.

***Digital Signature Standard (DSS)*:** Approaches to Digital Signatures (RSA Approach and DSA Approach), Digital Signature Algorithm.

Recommended Books:

- William Stallings; **Cryptography And Network Security: Principles And Practices** (Fourth Edition); Prentice Hall 2005
- Charles P. Pfleeger; **Security in Computing** (Fourth Edition); Pearson Education
- Wenbo Mao; **Modern Cryptography: Theory and Practice**; Prentice Hall

Further readings:

- William Stallings; **Network Security Essentials: Applications and Standards**; Prentice Hall

MJ–19 (Pr): Information Security Lab

1 Credit | 30 Minimum Class Hours | Semester VII

Objective:

The objective of the course is to enable students to–

- Become familiar with understand clipping techniques.
- Learn fundamentals of cryptography and its applications.
- Understand the concepts symmetric and asymmetric Cryptographic Techniques.

Learning Outcome:

After completion of this course, a student will be able to–

- Develop security mechanisms to protect computer systems and networks.
- Write programs that are more secure.
- Use cryptography algorithms and protocols to achieve computer security.

Experiment List

1. Write a program in C to implement Caesar Cipher encryption technique.
2. Write a program in C to implement Monoalphabetic Ciphers encryption technique.
3. Write a program in C to implement Playfair Cipher encryption technique.
4. Write a program in C to implement One-Time Pad encryption technique.
5. Write a program in C to implement simple RSA algorithm with small numbers.
6. Write a program in C to implement simplified DES.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *

Semester – VIII

Programme Structure for Four Year Undergraduate Programme (FYUGP) of Bachelor of Computer Applications (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
VIII	MN–2D	Minor From Vocational Studies/Discipline – 2 *(To be selected by the students from the list of available options)*		4	
	MJ–20	(Th): Introduction to Data Science	3–0–1	3	45
		(Pr): Data Science Lab		1	30
	AMJ–1	(Th): Artificial Intelligence	3–1–0	4	60
	AMJ–2	(Th): Assembly Language Programming	3–1–0	4	60
	AMJ–3	(Pr): Artificial Intelligence and Assembly Language Programming Lab	0–0–4	4	120
			Total Credits	**20**	

Abbreviations:
Th (Theory), **Pr** (Practical), **L–T–P** (Lecture–Tutorial–Practical), **MN–2** (Minor From Discipline–1, **MN–2** (Minor From Vocational Studies/Discipline–2), **MJ** (Major Disciplinary/Interdisciplinary Courses), **AMJ** (Advance Major Disciplinary/ Interdisciplinary Courses).

MJ–20 (Th): Introduction to Data Science

3 Credits | 45 Minimum Class Hours | Semester VIII

Objectives:

The key objective of data science is to extract valuable information for use in strategic decision making, product development, trends analysis, and forecasting. The key techniques in use are data mining, big data analysis, data extraction and data retrieval.

The main objectives of the course are as follows–

- To understand the overview and definition of Data Science with its crucial role in current business world.
- To understand the importance of mathematics & Statistics in Data Science.
- To understand the role of machine learning techniques in Data Science and its different types.
- To know the integrated role of computers and its components in Data Science.
- To understand the flow and process model of data science project management.

Learning Outcomes:

After completion of this course, a student will be able to–

- Students will be able to appreciate the need of data science in day to day life.
- They will be able to understand the process and components of data science project.
- Student will learn importance of probability and statistics in data science.
- Student will be able to understand the machine learning in today’s business world.
- Understands the various components of computer science being used for data science.
- Student will bale to understand the execution flow of a data science project

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination														
● Experiments – 15 Marks ● Viva-Voce – 05 Marks ● Practical File – 05 Marks														

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -‘B’		
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions	1 × 5	5

Group -'B'			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Data Science - An Overview	15
II	Mathematics and Statistics in Data Science	15
III	Machine Learning in Data Science	15
	Total	**45**

Detailed Syllabus

Unit I: Data Science - An Overview (15 Hours)

Introduction to Data Science, Terminologies related with Data Science, Basic framework and architecture, difference between Data Science and business analytics, Importance of Data Science in today's business world, Primary components of Data Science, Overview of different Data Science techniques, Challenges and opportunities in business analytics, Different industrial application of Data Science techniques.

Role of computer science in Data Science, Various components of computer science being used for Data Science, Role of relational data base systems in Data Science: SQL, NoSQL, Various freely available software tools used in Data Science: R, Python.

Unit II: Mathematics and Statistics in Data Science (15 Hours)

Role of mathematics in Data Science, Importance of probability and statistics in Data Science, Important types of statistical measures in Data Science : Descriptive, Predictive and Prescriptive statistics, Introduction to statistical inference and its usage in Data Science, Application of statistical techniques in Data Science, Overview of linear algebra : Matrix and Vector theory, Role of linear algebra in Data Science, Exploratory data analysis and visualization techniques, Difference between exploratory and descriptive statistics, EDA and Visualization as key component of Data Science.

Unit III: Machine Learning in Data Science (15 Hours)

Role of machine learning in Data Science, Different types of machine learning techniques and its broad scope in Data Science: Supervised, unsupervised, reinforcement and deep learning, Difference between different machine learning techniques, Brief introduction to machine learning algorithms, Importance of machine learning in today's business, Difference between machine learning classification and prediction.

Recommended Books:

- John W. Foreman; **Data Smart: Using Data Science to Transform Information into Insight** (1st Edition); Wiley Publication, 2015
- Joel Grus; **Data Science from Scratch: First Principles with Python** (1st Edition)

Further Readings:

- Lillian Pierson; **Data Science For Dummies**; 2015

MJ–20 (Th): Data Science Lab

1 Credit | 30 Minimum Class Hours | Semester VIII

Objectives:

The main objectives of the course are as follows–

- To understand the python libraries for data science
- To understand the basic Statistical and Probability measures for data science.
- To apply correlation and regression analytics on standard data sets.
- To present and interpret data using visualization packages in Python.

Learning Outcomes:

After completion of this course, a student will be able to–

- Make use of the python libraries for data science.
- Make use of the basic Statistical and Probability measures for data science.
- Perform correlation and regression analytics on standard data sets.
- Present and interpret data using visualization packages in Python.

Experiment List

1. Download, install and explore the features of NumPy, SciPy, Jupyter, and Pandas packages.

2. Working with NumPy arrays.

3. Reading data from text files, Excel and the web and exploring various commands for doing descriptive analytics on the Iris data set.

4. Use any data set (e.g., UCI and Pima Indians Diabetes data set) for performing the following:
    - Univariate analysis: Frequency, Mean, Median, Mode, Variance, Standard Deviation, Skewness and Kurtosis.
    - Bivariate analysis: Linear and logistic regression modeling.
    - Multiple Regression analysis.
    - Also compare the results of the above analysis for the two data sets.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

◆

AMJ–1 (Th): Artificial Intelligence

4 Credits | 60 Minimum Class Hours | Semester VIII

Objective:

The objective of the course is to enable students to–

- Understand the history, goal, basic concept, application, impact of AI.
- Learn the informed and uninformed search methods.
- Understand the fundamentals of knowledge representation.
- Understand important Machine Learning algorithms used for improving various components of an AI agent.

Learning Outcomes:

After completion of this course, a student will be able to–

- Demonstrate an understanding of the history, goal, basic concept, application, impact of AI and its foundations.
- Apply basic principles of AI in problem–solving that require perception, knowledge representation, inference, and learning.
- Demonstrate awareness and a fundamental understanding of various applications of AI and Machine Learning techniques in real–world problem solving.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=100

Semester Examination and Distribution of Marks

End Semester Examination (ESE) : 75 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -‘B’			
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

Semester Internal Examination (SIE) : 25 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
Group -‘B’			
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

Unit	Topic	Minimum Class Hours
I	Overview of Artificial Intelligence	10
II	Knowledge Representation, Reasoning, and Machine Learning	16
III	Artificial Neural Network (ANN)	17
IV	Natural Language Processing (NLP)	17
	Total	**60**

Detailed Syllabus

Unit I: Overview of Artificial Intelligence (10 Hours)

Introduction to Artificial Intelligence: Definition and History of AI, Basic Concepts and Goals of AI, Applications and Impact of AI in Various Fields.

Problem-Solving and Search Algorithms: Problem–Solving Methods in AI, Search Algorithms {Uninformed Search (Breadth First Search, Depth First Search, Uniform Cost Search), Informed Search (Greedy Search, A* Search)}.

Unit II: Knowledge Representation, Reasoning, and Machine Learning (ML) (16 Hours)

Knowledge Representation and Reasoning**:** Representing Knowledge in AI Systems, Propositional and Predicate Logic, Inference Rules and Reasoning Techniques.

Machine Learning**:** Introduction to ML, Supervised Learning, Unsupervised Learning, and Reinforcement Learning, Evaluation Metrics in ML.

Unit III: Artificial Neural Network (ANN) (17 Hours)

Introduction to ANN, Core Components of Neural Network Architecture (Input, Weight, Summation Function, Activation Function, Bias, Artificial Neuron, Layers), Working of ANN, Types of ANN (Feed Forward, Feedback, Single–Layer Perceptron, Multi–Layer Perceptron, Radial Basis Function, Convolutional Neural Network, Recurrent Neural Network), Characteristics of ANN, Properties of ANN, Advantages and Disadvantages of ANN, Applications of ANN.

Unit IV: Natural Language Processing (NLP) (17 Hours)

Introduction to NLP, Text Preprocessing Techniques (Regular Expression, Tokenization, Lemmatization and Stemming, Parts of Speech), Information Extraction Techniques (Named Entity Recognition, Relation Extraction, Event Extraction, Coreference Resolution, Template Filling, Open Information Extraction), Feature Extraction Techniques (One Hot Encoding, Bag of Word, n–grams, Tf–Idf, Custom Features, Word2Vec), Part Of Speech Tagging, Sentiment Analysis, Statistical Machine Translation of Languages, Text Summarization, Text Generation.

Recommended Books:

- Elaine Rich, Kevin Knight; **Artificial Intelligent**; Mc–Graw Hill
- Dan W. Patterson; **Introduction to AI and Expert System**; PHI

Further readings:

- Luger; **Artificial Intelligent**; Pearson Education
- Russel and Norvig; **Artificial Intelligent: A Modern Approach**; Pearson Education

AMJ–2 (Th): Assembly Language Programming

4 Credits | 60 Minimum Class Hours | Semester VIII

Objective:

The objective of the course is to–

- Be familiar with Assembly Language Programming.
- Study machine addressing, stack operations, subroutines, and programmed and interrupt driven I/O.
- Utilize the Intel x86 Architecture processor instruction set and to perform programming exercises.

Learning Outcomes:

After completion of this course, a student will be able to–

- Understand the underlying principle behind machine–level data representations, computing, and programming.
- Understand the underlying principles behind compiling, linking, loading, and debugging of Assembly Language Programming.
- Proficient with assembly programming for Intel x86 Architecture, including register operations, control structures, bitwise operations, and modular programs.

Outline of the Course

Minimum Class Hours		**Exam Time (Hours)**		**Credits**		**Marks**								
						Semester Internal		**End Semester**		**Full Mark**		**Pass Marks**		**Total Marks**
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=100

Semester Examination and Distribution of Marks

End Semester Examination (ESE) : 75 Marks

S.No	**Questions**		**Marks**
Group -‘A’			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -‘B’			
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

Semester Internal Examination (SIE) : 25 Marks

S.No	**Questions**		**Marks**
Group -‘A’			
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
Group -‘B’			
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

Unit	Topic	Minimum Class Hours
I	Assembly Language Fundamentals	20
II	Data Transfers, Instructions, Operators, and Addressing	20
III	Conditional Processing and Procedures	20
	Total	**60**

Detailed Syllabus

Unit I: Assembly Language Fundamentals (20 Hours)

Introduction to Assembly Language, Need of Assembly Language, Use of Assembly Language, Advantages and Disadvantages of Assembly Language, Assembly Language Program Development Tools (Editor, Assembler, Linker, Loader or Locator, Debugger, and Emulator).

Basic Elements of Assembly Language: Reserved Words, Identifiers, Constants, Directives, Instructions, The NOP (No Operation) Instruction, Template of Assembly Programming, Assembly Program Execution Using An Assembler Tool.

Defining Data: Intrinsic Data Types, Data Definition Statement, Defining Real Number Data, Little Endian Order, Addition of Two Variables, Declaring Uninitialized Data.

Unit II: Data Transfers, Instructions, Operators, and Addressing (20 Hours)

Data Transfer Instructions: Introduction, Operand Types, Direct Memory Operands, MOV Instruction, Zero/Sign Extension of Integers, LAHF and SAHF Instructions.

Arithmetic Instructions: INC, DEC, ADD, SUB, NEG, MUL, DIV, MOD.

Boolean and Comparison Instructions: The CPU Flags, AND, OR, XOR, NOT, TEST, CMP.

Shift and Rotate Instructions: Logical Shifts and Arithmetic Shifts, SHL Instruction, SHR Instruction, SAL and SAR Instructions.

Data–Related Operators and Directives: OFFSET Operator, ALIGN Directive, PTR Operator, TYPE Operator, LENGTHOF Operator, SIZEOF Operator, LABEL Directive.

Indirect Addressing: Indirect Operands, One–Dimensional Arrays, Indexed Operands.

Unit III: Conditional Processing and Procedures (20 Hours)

Jump Instruction: Unconditional Jump (JMP) Instruction, Conditional Structures, J<condition> Instruction.

Loop Instruction: LOOP Instruction, Conditional Loop Instructions.

Conditional Structures: Block–Structured IF Statements, WHILE Loop.

Defining and Using Procedures: PROC Directive, CALL and RET Instructions, Saving and Restoring Registers.

Recommended Books:

- Kip R. Irvine; **Assembly Language for x86 Processors** (Sixth Edition); Prentice Hall

Further readings:

- Randall Hyde; **The Art of Assembly Language** (Second Edition); No Starch Press, Inc., San Francisco

* * *

AMJ–3 (Pr): Artificial Intelligence and Assembly Language Programming Lab

4 Credit | 120 Minimum Class Hours | Semester VIII

Objective:

The objective of the course is to enable students to–

- Understand the history, goal, basic concept, application, impact of AI.
- Learn the informed and uninformed search methods.
- Understand the fundamentals of knowledge representation.
- Understand important Machine Learning algorithms used for improving various components of an AI agent.
- Be familiar with Assembly Language Programming.
- Study machine addressing, stack operations, subroutines, and programmed and interrupt driven I/O.
- Utilize the Intel x86 Architecture processor instruction set and to perform programming exercises.

Learning Outcomes:

After completion of this course, a student will be able to–

- Demonstrate an understanding of the history, goal, basic concept, application, impact of AI and its foundations.
- Apply basic principles of AI in problem–solving that require perception, knowledge representation, inference, and learning.
- Demonstrate awareness and a fundamental understanding of various applications of AI and Machine Learning techniques in real–world problem solving.
- Understand the underlying principle behind machine–level data representations, computing, and programming.
- Understand the underlying principles behind compiling, linking, loading, and debugging of Assembly Language Programming.
- Proficient with assembly programming for Intel x86 Architecture, including register operations, control structures, bitwise operations, and modular programs.
- Write Assembly Language to create both system level software tools and application programs.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
N/A	120	N/A	3	N/A	4	N/A	N/A	N/A	100	N/A	100	N/A	40	N/A+100=100

Marks Distribution of End Semester Practical Examination

● Experiments – 60 Marks ● Viva-Voce – 20 Marks ● Practical File – 20 Marks

Experiment List

Group – ‘A’: Artificial Intelligence

1. Write a program in Python to implement Breadth First Search.

2. Write a program in Python to implement Depth First Search.

3. Write a program in Python to implement Tic–Tac–Toe game.

4. Write a program in Python to implement 8–Puzzle problem.

5. Write a program in Python to implement Water–Jug problem.

6. Write a program in Python to implement A* Algorithm for a two–dimensional grid. Find shortest path between two nodes on the grid.

7. Write a program in Python to implement K–Nearest Neighbor Algorithm for data classification, choose dataset of your own choice.

8. Write a program in Python to implement Naive Bayes Algorithm for data classification, choose dataset of your own choice.

9. Write a program in Python to implement Logistic Regression for data classification, choose dataset of your own choice.

10. Write a program in Python to implement a simple rule–based system for medical diagnosis that can diagnose a couple diseases.

11. Write a program in Python to visualize popular Activation Functions used in Artificial Neural Network (ANN), aiding in understanding their behavior and suitability for different tasks.

12. Write a program in Python to recognize even and odd numbers represented in ASCII form, illustrating the basic principles of a Single–Layer Perceptron.

13. Write a program in Python to implement the training process of an Artificial Neural Network (ANN), covering the fundamental steps (i.e., Forward Propagation and Backward Propagation) in ANN.

14. Split the sentence “the quick brown fox, jumped over the lazy; dog” using .split() method on spaces and NLTK’s word_tokenize.

15. Write a program in Python with TensorFlow to classify a simple text based on **AG_NEWS** dataset, which is to classify news headlines into one of 4 categories: World, Sports, Business and Sci/Tech.

Group – ‘B’: Assembly Language Programming

Unit I: Assembly Language Fundamentals

16. Explain step by step procedure to install and Configure of an Assembler Tool (Either Microsoft Visual Studio or MASM or TASM or NASM or GAS or FASM or WinAsm or RadASM).

17. Write a program in Assembly Language to display “I am studying in <Your college name>”.

18. Write a program in Assembly Language to evaluate the arithmetic expression ((A + B / B * D - E) * (F – G)). Read the values of A, B, C, D, E, F, and G from the standard input device.

19. Write a program in Assembly Language to convert an ASCII code to its BCD equivalent.

Unit II: Data Transfers, Instructions, Operators, and Addressing

1. Write a program in Assembly Language to find the largest number among three numbers.

2. Write a program in Assembly Language to exchange two memory variables using MOV and XCHG instructions.

3. Write a program in Assembly Language to shift the value 10001111b once to the left.

Unit III: Conditional Processing

1. Write a program in Assembly Language to find factorial of a given number.

2. Write a program in Assembly Language to generate Fibonacci series up to Nth term.

3. Write a program in Assembly Language to find the HCF (or, GCD) and LCM of two 16–bit unsigned integers.

4. Write a program in Assembly Language to find the sum of all elements, average of all elements, and the second largest integer in a “One–Dimensional” integer array.

5. Write a program in Assembly Language to ask the user to enter his/her mobile number and calculate the sum of last four digits of his/her mobile number.

6. Write a program in Assembly Language that use procedure to return the HCF or GCD of two given 16–bit unsigned integers.

7. Write a procedure in Assembly Language to reverse the digits of a given number and test the procedure.

8. Write a program in Assembly Language that calculates the Nth Fibonacci series using recursive procedure.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

KOLHAN UNIVERSITY

Chaibasa, Jharkhand, India

Syllabus for
Four Year Undergraduate Programme (FYUGP)
of

Bachelor of Computer Application (BCA)

Minor From Discipline–1

Minor From Vocational Studies/Discipline – 2

With Effect From
Academic Year 2022 - 2023

As Per Revised Curriculum and Credit Framework for the FYUGP under the provisions of NEP - 2020

Table of Content

Programme Structure for
Four Year Undergraduate Programme (FYUGP)
of
Minor Courses of Bachelor of Computer Application (BCA)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
I	MN–1A	(Th): Introduction to Computer Science	3–0–1	3	45
		(Pr): Introduction to Computer Science Lab		1	30
II	MN–2A	(Th): Entrepreneurship Development	3–0–1	3	45
		(Pr): Entrepreneurship Development Lab		1	30
III	MN–1B	(Th): Fundaments of Digital Electronics	3–1–0	4	60
IV	MN–2B	(Th): Web Designing	3–0–1	3	45
		(Pr): Web Designing Lab		1	30
V	MN–1C	(Th): Electronic Commerce and Application	3–1–0	4	60
VI	MN–2C	(Th): Digital Marketing	3–0–1	3	45
		(Pr): Digital Marketing Lab		1	30
VII	MN–1D	(Th): Management Information System	3–1–0	4	60
VIII	MN–2D	(Th): Cyber Security	3–0–1	3	45
		(Pr): Cyber Security Lab		1	30
Total Credits				**32**	

Abbreviations:
L–T–P (Lecture–Tutorial–Practical), **MN-1** (Minor From Discipline-1), **MN-2** (Minor From Vocational Studies/ Discipline-2

Programme Objectives, Outcomes and Programme Specific Outcomes
of
Bachelor of Computer Application (BCA)

Programme Objectives:

1. **Foundational Knowledge in Programming and Development:**
    - Equip students with a strong foundation in programming languages such as C, C++, Java, Python, and Assembly Language, enabling them to develop efficient algorithms and software solutions.
    - Foster an understanding of Object-Oriented Programming and Data Structures for solving real-world computational problems.
2. **Mastery of Database and Data Management Concepts:**
    - Provide comprehensive knowledge of Relational Database Management Systems (RDBMS) and SQL for effective data organization, retrieval, and management.
3. **Proficiency in Software Engineering and System Analysis:**
    - Train students in Software Development Life Cycle (SDLC) methodologies, project management, and system analysis for designing scalable and maintainable software solutions.
4. **Understanding of Computer Systems and Networks:**
    - Develop a thorough understanding of Operating Systems, Computer Architecture, Digital Logic Design, and Data Communication Networks to manage system resources and ensure robust communication infrastructure.
5. **Web and Application Development Skills:**
    - Enable students to build dynamic web applications using technologies such as JSP, Web Technologies, and Python.
6. **Focus on Emerging Technologies:**
    - Introduce students to cutting-edge fields such as Data Science, Big Data Analytics, Artificial Intelligence, and Information Security, preparing them for future technological advancements.
7. **Enhancement of Analytical and Problem-Solving Skills:**
    - Strengthen computational thinking and problem-solving abilities through courses in Mathematical and Computational Analysis and Computer Oriented Numerical Analysis.
8. **Interdisciplinary Learning and Communication Skills:**
    - Promote interdisciplinary knowledge through Multi-Disciplinary Courses and enhance communication skills with Language and Communication Skills modules.

Programme Outcomes:

1. Acquire knowledge of Computer application theory and algorithm principles in the design and modeling of computer based system.

2. Understand the computing concepts and their applications using the acquired board based knowledge.

3. To provide thorough understanding of nature, scope and application of computer and computer languages.

4. Identify and analyze software application problems in multiple aspect including coding, testing and implementation in industrial applications.

5. The program prepares the young professional for a range of computer applications, computer organization, and techniques of Computer Networking, Software Engineering, Web development, Database management and Advance Java.

Programme Specific Outcomes:

1. To pursue further studies to get specialization in Computer Science and Applications, Economics, Mathematics, Business Administration.

2. To pursue the career in corporate sector can opt for MBA, MCA, etc.

3. To Work in the IT sector as programmer, system engineer, software tester, junior programmer, web developer, system administrator, software developer, etc.

4. To work in public sector undertakings and Government organizations.

5. Ability to understand the changes or future trends in the field of computer application.

6. Encouraging students to convert their start-up idea to reality by implementing.

7. Students will able to understand, analyze and develop computer programs in the areas related to algorithm, system software, web design and networking for efficient design of computer–based system.

Prepared and Proposed by:

Mr. Riki Sutradhar	**Mr. Subodh Kumar**
Assistant Professor (Computer Application) Jamshedpur Co-operative College, Jamshedpur	Assistant Professor (Computer Application) Jamshedpur Co-operative College, Jamshedpur

Programme Structure of Semester – I (Minor)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
I	MN–1A	(Th): Introduction to Computer Science	3–0–1	3	45
		(Pr): Introduction to Computer Science Lab		1	30

MN–1A (Th): Introduction to Computer Science

3 Credits | 45 Minimum Class Hours | Semester I

Objectives:

This course provides an overview of introductory concepts about computers, number systems and components of computer system. It builds the foundation of the computer application courses.

Learning Outcomes:

After completion of this course, a student will be able to–

- Handle a computer system for day-to-day use.
- Enumerate different types of input/output devices and types of memory.
- Perform conversion between different number systems including binary addition and subtraction.
- Familiarize Operating Systems, Programming languages, Network and Internet
- Differentiate between system and application software.
- Prepare documents, spreadsheets, and presentations.
- Use Email services.

Outline of the Course

Minimum Class Hours		**Exam Time (Hours)**		**Credits**		**Marks**									
						Semester Internal		**End Semester**		**Full Mark**		**Pass Marks**		**Total Marks**	
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**	
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100	
Marks Distribution of End Semester Practical Examination															
● Experiments – 15 Marks						● Viva-Voce – 05 Marks						● Practical File – 05 Marks			

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	**Questions**		**Marks**
Group -‘A’			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -‘B’			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	**Questions**		**Marks**
Group -‘A’			
1	Five very short answer type questions	1 × 5	5
Group -‘B’			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Computer Basics and Number System	12
II	Input/Output Devices	05
III	Computer Software and Programming Language	16
IV	Network and Internet	12
	Total	**45**

Detailed Syllabus

Unit I: Computer Basics and Number System (12 Hours)

Computer Basics: Introduction, Characteristics of Computers, Generation of Computers, Classification of Computers, Applications of Computers, Functional Block Diagram of Computer {Central Processing Unit, Input Unit, Output Unit, Memory Unit (Primary Storage Unit and Secondary Storage Unit), Bus Structure}.

Number System**:** Binary Number System, Octal Number System, Decimal Number System, Hexadecimal Number System, Conversion from One Number System to Another, Binary addition and subtraction.

Unit II: Input/Output Devices (5 Hours)

Introduction, Input Devices (Keyboard, Pointing Devices, Speech Recognition, Digital Camera, Scanners, Optical Scanners), Output Devices {Hard Copy Output Devices (Printers, Plotters, Computer Output Microfilm (COM)), Soft Copy Output Devices (Monitors, Speakers, Projectors}.

Unit III: Computer Software and Programming Language (16 Hours)

Computer Program**:** Introduction, Developing a Program, Algorithm, Flowchart, and Pseudo code.

Programming Language**:** Introduction, Evolution of Programming Languages, Classification of Programming Languages, Generations of Programming Languages, Features of a Good Programming Language, Selection of a Programming Language.

Computer Software**:** Introduction, Categories of Computer Software - System Software (Operating System, System Utility, Language Translators etc.) and Application Software (Word processor, Spreadsheet, Presentation, Database software etc.)

Unit IV: Network and Internet (12 Hours)

Introduction to Network, Types of Networks, Introduction to Internet, ISP, URL, IP Address, Web Page, Web Site, Web Server, Web Browser, Internet Services (WWW and Electronic Mail - Creating E-Mail account, sending and receiving E-Mails).

Recommended Books:

- Anita Goel; **Computer Fundamentals**; Pearson
- ITL Education Solution Limited, R&D Wing; **Introduction to Computer Science**; Pearson Education

Further readings:

- CI Stens School of Computing, **Internet and Introduction**, TMH
- Rajaraman V., **Fundamental of Computers**, Prentice Hall of India Pvt. Ltd., New Delhi
- Peter Nortons, **Introduction to Computer**, TMH

MN–1A (Pr): Introduction to Computer Science Lab

1 Credit | 30 Minimum Class Hours | Semester I

Objectives:

The main objectives of the course are as follows–

- To use standard word, spreadsheets, and presentation packages.
- To use Email services.
- To understanding computer hardware.
- To learning basic application software tools.

Learning Outcomes:

After completion of this course, a student will be able to–

- Handle a computer system for day-to-day including web browsing.
- Prepare documents, spreadsheets, and presentations.
- Use Email services.

Experiment List

	Unit I: Word Processor
1.	Prepare your Curriculum Vitae using various formatting tools.
2.	Using mail merge prepare an interview Call Letter.
3.	Create an index page for the Practical File using table having columns such as Sl. No, Experiment Name, Date of Experiment, Page No., Remarks etc. Use formatting tools, Table Layout option, Table Design options etc.
	Unit II: Spreadsheet
4.	Create a database of students, which contains marks obtained by students of a class in different subjects and then calculate maximum, minimum, average and sum of marks in each subject. Also calculate the percentage of total marks of each student using functions and formulas. Sort the data on the basis of percentage of total marks column in descending order.
5.	Draw effective charts (Pie Chart, Bar Graph etc.) to present students' data as given in Experiment No. 4.
	Unit III: Presentation
6.	Make a presentation on your college. Use various effects, animation and transitions etc. (Minimum 5 slides).
	Unit IV: Internet

7.	Create an e-mail id in any e-mail service provider website such as gmail, yahoomail, hotmail etc. **(i)** Write a mail to a software company for the post of Software Engineer. Send your resume as an attachment. Also send a copy of mail to your teacher as a blind carbon copy (BCC). **(ii)** Read the unread mails present in your mailbox. **(iii)** Delete the unnecessary mails from the inbox. **(iv)** Sign out from your email account.
8.	Search the official website of Kolhan University in Google search engine using any web browser. Open the official website of Kolhan University from Google search results and download the course **MN-1A (Introduction to Computer Science)** of FYUGP Syllabus.

Note: Additional lab assignments may be included based on topics covered in the theory paper

* * *

Programme Structure of Semester – II (Minor)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
II	MN–2A	(Th): Entrepreneurship Development	3–0–1	3	45
		(Pr): Entrepreneurship Development Lab		1	30

MN–2A (Th): Entrepreneurship Development

3 Credits | 45 Minimum Class Hours | Semester II

Objectives:

It provides exposure to the students to the entrepreneurial cultural and industrial growth so as to prepare them to set up and manage their own small units.

Learning Outcomes:

This course will be able to create value. Students are able to create presentations and business plans that articulate and apply financial, operational, organizational, market, and sales knowledge to identify paths to value creation through 1) company formation (for-profit); 2) social innovation (nonprofit).

Outline of the Course

Minimum Class Hours		**Exam Time (Hours)**		**Credits**		**Marks**								
						Semester Internal		**End Semester**		**Full Mark**		**Pass Marks**		**Total Marks**
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination														
● Experiments – 15 Marks					● Viva-Voce – 05 Marks					● Practical File – 05 Marks				

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	**Questions**		**Marks**
Group -'A'			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -'B'			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	**Questions**		**Marks**
Group -'A'			
1	Five very short answer type questions	1 × 5	5
Group -'B'			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	**Topic**	**Minimum Class Hours**
I	Introduction to the entrepreneur	08
II	Promotion of a Venture	08
III	Entrepreneurial Behaviour	10
IV	Entrepreneurship & Innovation	12
V	Legal and Ethical Considerations	07
	Total	**45**

Detailed Syllabus

Unit I: Introduction to the Entrepreneur (8 Hours)

Definition, Emergence of Entrepreneurial Class; Definition and Concern of Entrepreneurship, Role of Social Economic Environment; Classification, Characteristics and Importance of Entrepreneur; Leadership; Risk Taking; Decision Making and Business Planning, Role of Entrepreneur

Unit II: Promotion of a Venture (8 Hours)

Opportunities Analysis; External Environmental Analysis (Economic, Social and Technological, Competitive Factors), Legal Requirements of Establishment of a New Unit and Rising of Funds; Venture Capital Sources and Documentation Required.

Unit III: Entrepreneurial Behavior (10 Hours)

Innovation and Entrepreneur (Concept, Creativity, Invention and Innovation, Strategy for Innovation, Effective Commercialization, Innovation and Intellectual Property Rights), Entrepreneurial Behavior and Psycho–theories.

Entrepreneurial Development Programmes (EDP): EDP, Their Role, Relevance and Achievements; Role of Government in Organizing EDP's Critical Evaluation.

Unit IV: Entrepreneurship and Innovation (12 Hours)

Overview of Project Identification, Search of a Business Idea, Identification of Project, Business Opportunities, Understanding Design Thinking {Concept and Scope, Key Factors of Design Thinking, Benefits, Phases (Empathize, Define, Ideate, Prototype, Test)}

Creativity: Creativity, Identification Creative Tools (S–C–A–M–P–E–R), Vertical Thinking, Lateral Thinking, Critical Thinking, Phases of Decision Making, Critical Thinking and Objectivity, Applying Structured Knowledge to Unstructured Problems, Domain Criteria, Traditional and Out–of–the–box Thinking.

Unit V: Legal and Ethical Considerations (7 Hours)

Legal Forms of Business Organization, Ethical Issues and Social Responsibilities of an Entrepreneur.

Recommended Books:

- Vasant Desai; **Entrepreneurship**; Himalaya Publishing House, 2003
- Taneja, S.L. Gupta.; **Entrepreneurship Development**; 2003
- Pandey, I.M.; **Venture Capital – The Indian Experience**; Prentice Hall of India, 2003

Further readings:

- Tandon B.C.; **Environment and Entrepreneur**; Chug publication, Allahabad
- Srivastava S.B.; **A Practical Guide to Industrial Entrepreneurs**; Sultan Chand & Sons, New Delhi
- Chandra, Prasana; **Project Preparation, Appraisal, Implementation**; TMH, New Delhi

* * *

MN–2A (Pr): Entrepreneurship Development Lab

1 Credit | 30 Minimum Class Hours | Semester II

Objectives:

It provides exposure to the students to the entrepreneurial cultural and industrial growth so as to prepare them to set up and manage their own small units.

Learning Outcomes:

This course will be able to create value. Students are able to create presentations and business plans that articulate and apply financial, operational, organizational, market, and sales knowledge to identify paths to value creation through 1) company formation (for-profit); 2) social innovation (nonprofit).

Experiment List

1	Case studies of successful entrepreneurs.
2	Conducting mock interviews: testing initiatives, team spirit and leadership.
3	Conducting meeting: purpose, procedure, participation, physical arrangements, recording and writing of minutes.
4	Share Your Story: Identify area of innovation and prepare a project of design thinking in the area of Your choice and present it through Sketch modeling (Preparing project proposal).
5	Conduct Market survey to know the demand for different products.
6	Presentations by the students (Individuals, Group).

* * *

Programme Structure of Semester – III (Minor)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
III	MN–1B	(Th): Fundaments of Digital Electronics	3–1–0	4	60

MN–1B (Th): Fundamentals of Digital Electronics

4 Credits | 60 Minimum Class Hours | Semester III

Objective:

The objective of the course is to enable students to–

- Understand how to represent Binary, Octal, Decimal, and Hexadecimal data and perform the conversion among different number systems.
- Understand the application of Logic Circuit and Boolean algebra in Computer Science and Applications.
- Understand the design of various functional units and digital components of a computer.

Learning Outcome:

After completion of this course, a student will be able to–

- Analyze the performance of commercially available computers.
- Build simple logic circuits using basic logic gates such as AND, OR, NOT, NAND, and NOR.
- Understand the architecture of various digital components, like Integrated Circuits, Decoders, Encoders, Multiplexers, De–multiplexers, Registers, Shift Registers, and Binary Counters.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=100

Semester Examination and Distribution of Marks

End Semester Examination (ESE) : 75 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -'B'		
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

Semester Internal Examination (SIE) : 25 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
	Group -'B'		
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

Unit	Topic	Minimum Class Hours
I	Data Representation	10
II	Digital Logic Circuits	25
III	Digital Components	25
	Total	**60**

Detailed Syllabus

Unit I: Data Representation (10 Hours)

***Number System*:** Binary number system, Octal number system, Decimal number system, Hexadecimal number system, Conversion from one number system to another, Binary Arithmetic (Addition, Subtraction, Multiplication, and Division).

***Fixed–Point Number (i.e., Integer) Representation*:** Unsigned integers, Signed integers (Sign–magnitude, 1's complement, and 2's complement representation).

***Floating–Point Number Representation*:** 32–bit single–precision floating–point numbers, 64–bit double–precision floating–point numbers.

***Character Encoding*:** Bit, Byte, Word, BCD, EBCDIC, ASCII, ANSI, Unicode, UTF, ISCII.

Unit II: Digital Logic Circuits (25 Hours)

***Logic Circuit*:** Logic Gates (AND, OR, NOT, NAND, NOR, Exclusive–OR, Exclusive–NOR), Converting expressions to logic circuits.

***Boolean Algebra*:** Fundamental concepts of Boolean algebra, Postulates of Boolean algebra, Representation of Boolean expressions using truth tables, The principle of Duality/Perfect induction, De–Morgan's theorem, Simplification of Boolean expression, Canonical forms for Boolean expressions (Sum–of–Product and Product–of–Sum), Conversion between canonical forms.

***Combinational Circuits*:** Half adder, Full adder.

***Flip–Flops*:** Latches, Edge triggered flip–flops (SR flip–flops, D flip–flops, JK flip–flops, and T flip–flops), Pulse triggered flip–flops (Master slave JK flip–flop).

Unit III: Digital Components (25 Hours)

Integrated Circuits (Types of Integrated Circuits Based on Number of Gates, Types of Integrated Circuits Based on Circuit Technology), Decoders, Encoders, Multiplexers, De–multiplexers, Registers {Modes of Operation (SISO, SIPO, PISO and PIPO)}, Shift Registers, Binary Counters {Asynchronous counters (Four bit ripple counter), Synchronous counter (Four bit synchronous counter)}.

Recommended Books:

- M. Morris Mano; **Computer System Architecture** (Third Edition); New Delhi: Prentice-Hall India, 2002
- Donald P Leach, Albert Paul Malvino, Goutam Saha; **Digital Principles and Applications** (Seventh Edition); New Delhi: Tata McGraw Hill Education Pvt. Ltd., 2011
- Mostafa Abd–El–Barr, Hesham El–Rewini; **Fundamentals of Computer Organization and Architecture**; John Willy and Sons, Inc. Publication, 2005
- Thomas L. Floyd; **Digital Fundamentals** (Fifth Edition); New Delhi: Pearson Education, 2002

Further readings:

- William Stallings; **Computer Organization and Architecture** (Sixth Edition); New Delhi: Prentice-Hall India, 2002
- B. Ram, Sanjay Kumar; **Computer Fundamentals: Architecture and Organization** (Fifth Edition); New Age International Pvt. Ltd.. 2018

* * *

Programme Structure of Semester – IV (Minor)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
IV	MN–2B	(Th): Web Designing	3–0–1	3	45
		(Pr): Web Designing Lab		1	30

MN–2B (Th): Web Designing

3 Credits | 45 Minimum Class Hours | Semester IV

Objective:

The objective of the course is to familiarize the students with a discussion on internet and its growth. It also provides the students a study on the basic services provided by internet and a familiarization on the markup languages and CSS. Static web page development is also being discussed to make the student competent to design static websites.

Learning Outcome:

After completion of this course, a student will be able to–

- Gain a good grounding of Web Application Terminologies, Internet Tools, E–Commerce and other web services.
- Design and implement static websites with good aesthetic sense of designing using HTML and CSS.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination														
● Experiments – 15 Marks					● Viva-Voce – 05 Marks					● Practical File – 05 Marks				

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -‘B’			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
Group -‘A’			
1	Five very short answer type questions	1 × 5	5
Group -‘B’			
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Introduction to Web Technology	15
II	Introduction to Markup Languages	20
III	Web Page Designing With CSS	10
	Total	**45**

Detailed Syllabus

Unit I: Introduction to Web Technology (15 Hours)

***Internet*:** Introduction to Internet, Introduction to World Wide Web (WWW), W3C (World Wide Web Consortium), Levels of Internet Connectivity (Dial–up, Leased Line, DSL, VSAT), Requirements for Internet Connectivity, Web Protocols (TCP/IP, UDP/IP, HTTP, HTTPS, FTP, DHCP), Electronic Mail, Internet Relay Chat, Instant Messaging, Search Engines, URL, DNS, Proxy Servers, Internet Service Provider.

Web Basics: Basics of Web Browsers (Internet Explorer, Google Chrome, Mozilla Firefox, Safari, Opera, Netscape Navigator, Konqueror, Lynx), Basics of Web Servers {Apache HTTP Server, Apache Tomcat, Microsoft's Internet Information Services (IIS), Nginx Web Server, Lighttpd, Jigsaw, Klone, Abyss Web Server, Oracle Web Tier, X5 (Xitami) Web Server, Zeus Web Server}, Web Pages (Static, Dynamic, Active), Web Hosting and DNS.

Unit II: Introduction to Markup Languages (20 Hours)

Definition of GML, SGML, VRML, UML, XML, and XHTML.

***HTML Basic*:** Introduction to HTML, The Structure of HTML Document, Global Attributes, Document Structure Tags, Unclosed Tags, Text Formatting Tags, Link Tags, List Tags, Image and Object Tags, Executable Content Tags, Some More Tags (DIV, MARQUEE, NOBR, DFN, HR, COMMENT), HTML Fonts, HTML Color, HTML Elements (HTML Tag vs. HTML Element, Nested HTML Elements), HTML Attributes (Core Attributes, Internationalization Attributes).

***Tables*:** Introduction to HTML Tables, Table Tags, Alignment, Aligning Entire Table, Alignment within a Row, Alignment within a Cell, Attributes, Content Summary, Background Color, Adding a Caption, Setting the Width, Adding a Border, Spacing Within a Cells, Spanning Multiple Rows or Columns, Table Sections and Column Properties, Table as a Design Tool.

***Frames*:** Introduction to Frames and Its Applications, Frames Document, The FRAMESET Tag, Nesting FRAMESET Tag, Placing Content in Frames With The FRAME Tag, Targeting Named Frames, Creating Floating Frames, Using Hidden Frames.

***Forms*:** Creating Forms, The FORM Tag, Named Input Fields, The INPUT Tag, Multiple Lines Text Windows, Drop Down and List Boxes, Hidden Text, Password, File Upload, Button, Submit, Reset, Radio, Checkbox, Select, Option, Forms and Scripting, Action Buttons, Labeling Input Files, Grouping Related Fields, Disabled and Read–only Fields, Form Field Event Handlers, Passing Form Data.

Unit III: Cascading Style Sheet (10 Hours)

Introduction to CSS, CSS Syntax, CSS Selector, How to Add CSS (Inline, Internal, External), CSS Comments, CSS Properties (Background, Border, Display, Float, Font, Line Height, Margin, Opacity, Overflow, Padding, Position, Vertical Align, White Space, Width, Word Wrap, Outline, Visibility, Counter).

Recommended Books:

- Harvey M. Deitel, Paul J. Deitel, Abbey Deitel; **Internet and World Wide Web: How to Program** (Fifth Edition); Pearson Education, 2012
- Thomas A. Powell, **HTML & CSS: The Complete Reference** (Fifth Edition); McGraw Hill, 2010

Further readings:

- Thomas A. Powell; **Web Design: The Complete Reference** (Second Edition); Tata McGraw Hill, 2002

MN–2B (Pr): Web Designing Lab

1 Credit | 30 Minimum Class Hours | Semester IV

Objective:

The objective of the course is to familiarize the students with the markup languages, scripting languages, and CSS. Static web page development is also being discussed to make the student competent to design static websites.

Learning Outcome:

After completion of this course, a student will be able to design and implement static websites with good aesthetic sense of designing using HTML and CSS.

Experiment List

Unit I: Introduction to Markup Languages

1. Create a static website for your family. The home page should contain a suitable name of your family, introduction, postal address, common mobile/land–line number and the picture of the family members. Different heading tags and paragraphs in different face, size, and color of fonts should be applied wherever necessary. Apply all necessary attributes of image tag to the picture of the family members. The names of each family member should be added at the bottom of corresponding picture of the family members, where each of the names should link to a personal page of the corresponding person, i.e., there should be separate personal page about each of the family members including personal information such as name, personal mobile number, e–mail id, date of birth, hobbies and favorite pastimes, etc. Also use the picture of the family members as image maps, where each of the pictures should link to the same personal page of the corresponding person.

2. Create a HTML document to display an ordered list of your top five favorite movies. Each element of the list must have a nested unordered list of songs with name of vocal artist. You may apply text formatting tags to the lists.

3. Design an image map of Indian flag with clickable areas of tricolors. Display respective color information in new HTML page when it is clicked.

4. Create a HTML document to display the Time–Table of your class. Use HTML table tags and the attributes (such as align, border, bgcolor, colspan, rowspan, cellspacing, cellpadding, background, width, and height) wherever necessary. A specimen of the Time–Table is given bellow–

TIME–TABLE

Day/Period	**I 09:30-10:20**	**II 10:20-11:10**	**III 11:10-12:00**	**12:00-12:40**	**IV 12:40-01:30**	**V 01:30-02:20**	**VI 02:20-03:10**	**VII 03:10-04:00**
Monday	Eng	Mat	Che	**LUNCH**	Lab			Phy
Tuesday	Lab				Eng	Che	Mat	Sports
Wednesday	Mat	Phy	Eng		Che	Library		
Thursday	Phy	Eng	Che		Lab			Mat
Friday	Lab				Mat	Che	Eng	Phy
Saturday	Eng	Che	Mat		Seminar			Sports

5. Create a website that displays the introduction to four courses, viz., Data Mining, Big Data Analytics, Artificial Intelligence, and Information Security in left panel. Add the name of institution with logo in header section, notice board in right panel, and date of creation with developer's name in footer section. After clicking link given in left panel, the appropriate content of that course must be displayed in place of default section. (Use FRAMESET and FRAME Tag). A specimen of the website is given bellow–

<table>
<tr><td colspan="3">Name of Institution with Logo</td></tr>
<tr><td><b><u>Data Mining</u></b>
<b><u>Big Data Analytics</u></b>
<b><u>Artificial Intelligence</u></b>
<b><u>Information Security</u></b></td><td><b>Default Section</b>
After clicking link given in left panel, the appropriate content of that course must be displayed in this section</td><td>News/Other Instructions in MARQUEE Effect</td></tr>
<tr><td>Date of Creation:
<Current Date></td><td colspan="2">Developed By <Your Name></td></tr>
</table>

6. Create a HTML page that displays a form with the information – First Name, Last Name, Gender, Subject of Interest, Language Known, Date of Birth (DD–MM–YYYY), Contact No., Address, Pin Code, State, and Country with Reset and Submit button in tabular format.

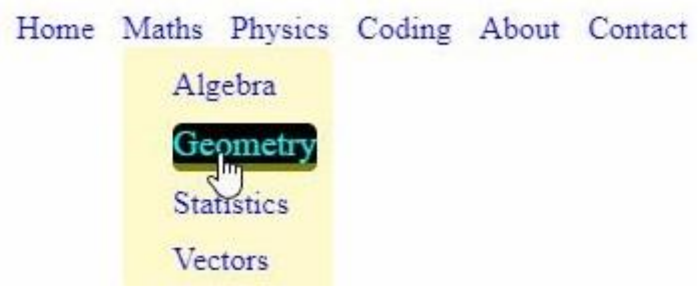
Unit II: Web Page Designing With CSS

1. Create an external CSS named "MyForm.css" with the following style formats–

- For div tag – border-radius: 5px; background-color: #f2f2f2; padding: 20px;
- For h1 header – font face: Bookman Old Style, Size: 24pts, color: green;
- For input tag – width: 100%; padding: 12px 20px; margin: 8px 0; display: inline-block; border: 1px solid #ccc; border-radius: 4px; box-sizing: border-box;
- For Reset and Register button – width: 100%; background-color: #4CAF50; color: white; padding: 14px 20px; margin: 8px 0; border: none; border-radius: 4px; cursor: pointer;

Design a web page that contains a div tag. Inside this div tag, place a registration form with the information – Name, Age, Username, Password, and Re–enter Password with Reset and Register button in tabular format to demonstrate the styles defined in the file "MyForm.css".

2. Design a web page that contains a drop–down menu using CSS as follows–



Note: Additional lab assignments may be included based on topics covered in the theory paper.

Programme Structure of Semester – V (Minor)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
V	MN–1C	(Th): E–Commerce and Application	3–1–0	4	60

MN–1C (Th): E–Commerce and Application

4 Credits | 60 Minimum Class Hours | Semester V

Objectives:

This course intends to make the students familiar with the required mechanisms for conducting business transactions through electronic means.

Learning Outcomes:

After completion of this course, a student will be able to–

- Understand the business, web technology and network concepts of electronic market in the fast changing business environment.
- Design and analyze the different models for the various E–Commerce companies based on the different business standards and payment methods.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=100

Semester Examination and Distribution of Marks

End Semester Examination (ESE) : 75 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -'B'			
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

Semester Internal Examination (SIE) : 25 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
Group -'B'			
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

Unit	Topic	Minimum Class Hours
I	Introduction to E-commerce	20
II	Building an E-Commerce Website	10
III	Security and Encryption	10
IV	Electronic Data Exchange and E-Commerce Payment Systems	20
	Total	**60**

Detailed Syllabus

Unit I: Introduction to E–Commerce (20 Hours)

What is E–Commerce (Introduction and Definition), Main activities E–Commerce, Goals of E–Commerce, Technical Components of E–Commerce, Functions of E–Commerce, Advantages and Disadvantages of E–Commerce, Scope of E–Commerce, Applications of E–Commerce, Electronic Commerce and Electronic Business (C2C, B2B, B2C, C2B, B2G or B2A, C2G or C2A), Mobile Commerce (M–Commerce).

Unit II: Building an E–Commerce Website (10 Hours)

A Systematic Approach, Choosing Server Software, Choosing the Hardware for an E–Commerce Site, Other E–Commerce Site Tools.

Unit III: Security and Encryption (10 Hours)

The E–Commerce Security Environment, Security Threats in the E–Commerce Environment, Technology Solutions, Policies, Procedures and Laws.

Unit IV: Electronic Data Interchange and E–Commerce Payment Systems (20 Hours)

Electronic Data Interchange: Introduction, Concepts of EDI and Limitation, Applications of EDI, Disadvantages of EDI, EDI Model.

Electronic Payment System: Introduction, Types of Electronic Payment System, Traditional Payment, Value Exchange System, Credit Card System, Electronic Fund Transfer, Paperless Bill, Modern Payment Cash, Electronic Cash (Digital Rupee or E–Rupee), Unified Payments Interface (UPI).

Recommended Books:

- K. K. Bajaj, D. Nag; **E–Commerce** (Second Edition); Tata McGraw Hill, New Delhi
- Henry Chan, Raymond Lee, Tharam Dillon, Elizabeth Chang; **E–Commerce: Fundamentals and Applications**; Wiley, India

Further readings:

- G. S. V. Murthy; **E–Commerce Concepts, Models, Strategies**; Himalaya Publishing House
- Kamlesh K Bajaj and Debjani Nag; **E–Commerce**
- Gray P. Schneider; **Electronic Commerce**

◆

Programme Structure of Semester – VI (Minor)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
VI	MN–2C	(Th): Digital Marketing	3–0–1	3	45
		(Pr): Digital Marketing Lab		1	30

MN–2C (Th): Digital Marketing

3 Credits | 45 Minimum Class Hours | Semester VI

Objective:

The objectives of the course are–

- Introduce learners to the core concepts, principles, and strategies of digital marketing.
- Teach learners how to plan, design, and structure an effective website aligned with marketing goals.
- Introduce learners to the role of social media platforms in digital marketing.
- Provide a comprehensive understanding of SEO principles and techniques to improve website visibility.

Learning Outcomes:

After completion of this course, a student will be able to–

- Define digital marketing and differentiate it from traditional marketing.
- Identify the primary components of a digital marketing strategy (e.g., content, channels, audience targeting).
- Develop a website plan that includes goals, target audience, and content structure.
- Conduct keyword research and select relevant keywords for a given business or campaign.
- Optimize website content, meta tags, and images for search engine visibility.
- Identify the appropriate social media platforms for specific target audiences and business goals.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100
Marks Distribution of End Semester Practical Examination ● Experiments – 15 Marks ● Viva-Voce – 05 Marks ● Practical File – 05 Marks														

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
Group -'B'			
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
Group -'A'			
1	Five very short answer type questions	1 × 5	5
Group -'B'			

3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Fundamentals of Digital Marketing	15
II	Website Planning & Analysis	10
III	Search Engine Optimization (SEO)	10
IV	Social Media Marketing	10
	Total	**45**

Detailed Syllabus

Unit I: Fundamentals of Digital Marketing (15 Hours)

Fundamentals of Digital marketing & Its Significance, Traditional marketing vs. Digital Marketing, Evolution of Digital Marketing, Terminology used in Digital Marketing, Understanding Digital Marketing Process, Visibility in Digital Marketing, Types of Visibility, Ways to Increase Visibility, Visitor Engagement, Inbound and Outbound Marketing, Converting Traffic into Leads, Types of Conversion, Understanding Conversion Process Tools Needed.

Unit II: Website Planning and Analysis (10 Hours)

Definition and Importance of Domain Names and Extensions, Identifying the Best Hosting Servers, Understanding Various Server Types, Utilizing a Content Management System. Popular User Experience Trends, Defining Website Objectives, Essential Steps in Website Creation.

Unit III: Search Engine Optimization (SEO) (10 Hours)

An Overview of SERP and SEO, How Search engine works, How SEO Works, Types of SEO technique, Different types of keywords, Strategies for keyword optimization, Content planning and optimization.

Unit IV: Social Media Marketing (10 Hours)

Fundamentals of Social Media Marketing and its Significance, Necessity of Social Media Marketing, Building a Successful Strategy (Goal Setting, Implementation), Instagram Marketing, X (Twitter) Marketing, Facebook Marketing, LinkedIn Marketing, YouTube Marketing.

Recommended Books:

- Kamat and Kamat; **Digital Marketing**; Himalaya
- V. Ahuja; **Digital Marketing**; Oxford University Press
- S. Gupta; **Digital Marketing**; McGraw–Hill

Further Readings:

- D. Ryan; **Marketing Strategies for Engaging the Digital Generation**

MN–2C (Pr): Digital Marketing Lab

1 Credits | 30 Minimum Class Hours | Semester VI

Objective:

The objectives of the course are–

- Understand visibility, visitor engagement, and the conversion process in digital marketing.
- Plan and analyze the foundational elements of a website.
- Apply SEO techniques to optimize content and analyze keyword performance.
- Design and evaluate a social media marketing strategy for multiple platforms.

Learning Outcomes:

After completion of this course, a student will be able to–

- Identify the primary components of a digital marketing strategy (e.g., content, channels, audience targeting).
- Develop a website plan that includes goals, target audience, and content structure.
- Conduct keyword research and select relevant keywords for a given business or campaign.
- Optimize website content, meta tags, and images for search engine visibility.
- Identify the appropriate social media platforms for specific target audiences and business goals.

Experiment List

Unit I: Digital Marketing Visibility and Conversion Analysis

Lab Assignment 1:

(i) Visibility Audit:

→ Choose a small business website or blog (e.g., a local store or personal project).

→ Identify at least 3 ways the website increases its visibility (e.g., blog posts, social media links, SEO keywords).

(ii) Engagement and Conversion Exercise:

→ Analyze a sample webpage (provide a URL or use a demo site).

→ Identify elements designed for visitor engagement (e.g., call-to-action buttons, forms).

→ Propose a simple inbound marketing tactic (e.g., a free eBook download) to convert traffic into leads.

Tools Needed: Google Analytics (demo account or free version), any website with public data (e.g., a blog).

A 2–page report summarizing the visibility audit, engagement analysis.

Unit II: Website Planning and Analysis

Lab Assignment 2:

(i) Domain and Hosting Research:

→ Research and list 3 domain name options for a hypothetical business (e.g., a bakery) with appropriate extensions (.com, .co, .shop).

→ Compare 2 hosting providers (e.g., Bluehost vs. HostGator) based on pricing, server types, and features.

(ii) Website Objectives & Structure:

→ Define the objectives for the hypothetical business website (e.g., sell products, build a mailing list).

→ Sketch a basic website structure (homepage, about page, contact page, etc.) using a flowchart or list.

Tools Needed: WordPress (free version or local setup), Google Docs, Google Analytics (optional).

A 1–2 page document with domain/hosting research, website objectives, and structure.

Unit III: Search Engine Optimization (SEO)

Lab Assignment 3:

(i) Keyword Research:

→ Choose a niche (e.g., fitness, travel, tech gadgets).

→ Use Google Keyword Planner or Ubersuggest to identify 5 high–volume, low–competition keywords.

→ Classify them into short-tail and long-tail keywords.

(ii) SEO Technique Application:

→ Suggest 2 on–page SEO techniques (e.g., alt text for images, internal linking) and 1 off–page technique (e.g., backlink outreach) for the blog post.

Tools Needed: Google Keyword Planner (free via Google Ads), Ubersuggest (free tier), any text editor.

A document containing the keyword list, and suggested SEO techniques.

Unit IV: Social Media Marketing

Lab Assignment 4:

(i) Content Creation:

→ Design a post for selected platform (e.g., an Instagram image, an X tweet, a LinkedIn article teaser).

→ Tailor the content to each platform’s audience and format (e.g., hashtags for Instagram, concise text for X).

(ii) Performance Metrics:

→ Create a simple tracking table in Google Sheets with columns: Platform, Post Type, Goal, Metrics (e.g., likes, shares, clicks).

→ Hypothesize expected results for each post (e.g., 50 likes on Instagram).

Tools Needed: Canva (free tier), Google Sheets, social media platforms (personal accounts or mockups).

A report with screenshots of the selected social media post and its descriptions.

A Google Sheet (or screenshot) with the tracking table and hypothesized metrics.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

Programme Structure of Semester – VII (Minor)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
VII	MN–1D	(Th): Management Information System	3–1–0	4	60

MN–1D (Th): Management Information System

4 Credits | 60 Minimum Class Hours | Semester VII

Objective:

The objectives of the course are–

- Apply sound management concepts and principles in the development and operation of information system.
- Apply system analysis, IS design and project management concepts effectively.
- Improve business processes through the effective application of information technology concepts and practices.
- Acquaint the students about the concept of information system in business organizations, and also the management control systems.

Learning Outcomes:

After completion of this course, a student will be able to–

- Analyze a complex computing problems and to apply principles of computing and other relevant disciplines to identify solutions.
- Design, implement and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
- Illustrate redesigning the organization with information system.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	N/A	3	N/A	4	N/A	25	N/A	75	N/A	100	N/A	40	N/A	100+N/A=100

Semester Examination and Distribution of Marks

End Semester Examination (ESE) : 75 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -‘B’		
3	Descriptive type question – four out of six	15 × 4	60
		Total Marks	**75**

Semester Internal Examination (SIE) : 25 Marks

S.No	Questions		Marks
	Group -‘A’		
1	Five very short answer type questions	1 × 5	5
2	One short answer type questions	5 × 1	5
	Group -‘B’		
3	Descriptive type question – one out of two	10 × 1	10
4	Attendance/ overall class performance in the semester		5
		Total Marks	**25**

Unit	Topic	Minimum Class Hours
I	Management Information System: A Framework	18
II	Information and System Concepts	6
III	System Design	18
IV	Implementation of MIS	8
V	Database Management System	10
	Total	**60**

Detailed Syllabus

Unit I: Management Information System: A Framework (18 Hours)

Definition, Purpose, Objectives and Role of MIS in Business Organization with particular reference to Management Levels. MIS Growth and Development, Location of MIS in the Organization concept and design. Transaction Processing System, Decision Support System, Executive Information system, Expert System, and the recent developments in the field of MIS. MIS and Decision Making Concepts.

Unit II: Information and System Concepts (6 Hours)

Concept of Information, Classification of Information, Value of Information, MIS and Information Concepts.

Unit III: System Design (18 Hours)

Concept of System, Types of Systems – Open, Closed, Deterministic, Probabilistic, etc, System Development Life Cycle Concept of Systems Analysis and Design (SAD), System development life cycle ,Need for Systems Analysis- SRS, DFD, DD & Decision tables, the Process of SAD, MIS and System Analysis, System Design – design methods, detailed system design, design documentation, System Implementation & testing.

Unit IV: Implementation of MIS (8 Hours)

Planning, Designing, and Implementation of MIS, Concept of Philosophy of DSS, Introduction to Enterprise Management System.

Unit V: Database Management System (10 Hours)

The Concept of DBMS, Objectives, role, advantages & disadvantages of DBMS, SQL, use of databases for integration across functional areas and RDBMS.

Recommended Books:

- D.P. Goyal; **Management Information Systems**; MacMillan
- Davis and Olson; **Management Information Systems**
- Brian O.; **Management Information System**; Tata McGraw Hill
- Murdick, R.G. and Ross J.E.; **Information System for Modern Management**; Tata PHI: New Delhi

References Books:

- Jawadekar; **Management Information System**; Tata McGraw Hill
- Jain Sarika; **Information System**; PPM
- Davis; **Information System**; Palgrave Macmillan

Programme Structure of Semester – VIII (Minor)

Sem.	Course Code	Course Title	L–T–P	Credits	Contact Hours
VIII	MN–2D	(Th): Cyber Security	3–0–1	3	45
		(Pr): Cyber Security Lab		1	30

MN–2D (Th): Cyber Security

3 Credits | 45 Minimum Class Hours | Semester VIII

Objective:

The objective of the course is to enable students to–

- Learn the foundations of cyber security and threat landscape.
- Equip students with the technical knowledge and skills needed to protect and defend against cyber threats.
- Develop skills in students that can help them plan, implement, and monitor cyber security mechanisms to ensure the protection of information technology assets.
- Expose students to responsible use of online social media networks.
- To systematically educate the necessity to understand the impact of cyber crimes and threats with solutions in a global and societal context.

Learning Outcome:

After completion of this course, a student will be able to–

- Understand the cyber security threat landscape.
- Develop a deeper understanding and familiarity with various types of cyber attacks, cyber crimes, vulnerabilities and remedies thereto.
- Analyze and evaluate existing legal framework and laws on cyber security.
- Analyze and evaluate the digital payment system security and remedial measures against digital payment frauds.
- Analyze and evaluate the importance of personal data its privacy and security.
- Analyze and evaluate the security aspects of social media platforms and ethical aspects associated with use of social media.

Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks								
						Semester Internal		End Semester		Full Mark		Pass Marks		Total Marks
Th	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th**	**Pr**	**Th + Pr**
45	30	3	3	3	1	15	N/A	60	25	75	25	30	10	75+25=100

Marks Distribution of End Semester Practical Examination

- Experiments – 15 Marks
- Viva-Voce – 05 Marks
- Practical File – 05 Marks

Semester Examination and Distribution of Marks

End Semester Examination (ESE): 60 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions (MCQ not to be set)	1 × 5	5
2	Two short answer type questions	5 × 2	10
	Group -'B'		
3	Descriptive type question – three out of five	15 × 3	45
		Total Marks	**60**

Semester Internal Examination (SIE): 15 Marks

S.No	Questions		Marks
	Group -'A'		
1	Five very short answer type questions	1 × 5	5
	Group -'B'		
3	Descriptive type question – one out of two	5 × 1	5
4	Attendance/ overall class performance in the semester		5
		Total Marks	**15**

Unit	Topic	Minimum Class Hours
I	Overview of Cyber Security	10
II	Overview of Social Media and Security	10
III	E–Commerce and Digital Payments	15
IV	Digital Devices Security, Tools and Technologies for Cyber Security	10
	Total	**45**

Detailed Syllabus

Unit I: Overview of Cyber Security (10 Hours)

Introduction to Cyberspace, Architecture of Cyberspace, Internet Infrastructure for Data Transfer and Governance, Regulation of Cyberspace, Concept of Cyber Security, Issues and Challenges of Cyber Security.

Unit II: Overview of Social Media and Security (10 Hours)

Introduction to Social Networks, Types of Social Media, Social Media platforms, Social Media Monitoring, Hashtag, Viral Content, Social Media Marketing, Social Media Privacy, Challenges, Opportunities and Pitfalls in Online Social Network, Security Issues Related to Social Media, Flagging and Reporting of Inappropriate Content, Laws Regarding Posting of Inappropriate Content, Best Practices for the Use of Social Media.

Unit III: E–Commerce and Digital Payments (15 Hours)

Elements of E–Commerce Security, E–Commerce Threats, Best Practices of E–Commerce Security, Introduction to Digital Payments, Components of Digital Payment and Stake Holders, Modes of Digital Payments {Banking Cards, Unified Payment Interface (UPI), e–Wallets, Unstructured Supplementary Service Data (USSD), Aadhaar Enabled Payments}, Common Frauds and Preventive Measures Related to Digital Payments. RBI Guidelines on Digital Payments and Customer Protection in Unauthorized Banking Transactions.

Unit IV: Digital Devices Security, Tools and Technologies for Cyber Security (10 Hours)

End Point Device and Mobile Phone Security, Password Policy, Security Patch Management, Data Backup, Downloading and Management of Third Party Software, Device Security Policy, Cyber Security Best Practices, Significance and Management of Host Firewall and Ant–Virus, Wi–Fi Security, Configuration of Basic Security Policy and Permissions.

Recommended Books:

- R. C. Mishra; **Cyber Crime Impact in the New Millennium**; Auther Press; 2010

- Sumit Belapure, Nina Godbole; **Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives**; Wiley India Pvt. Ltd. (First Edition, 2011)
- Henry A. Oliver; **Security in the Digital Age: Social Media Security Threats and Vulnerabilities**; Create Space Independent Publishing Platform. (Pearson , 13th November, 2001)
- Elias M. Awad; **Electronic Commerce**; Prentice Hall of India Pvt. Ltd.
- Kumar K.; **Cyber Laws: Intellectual Property & E–Commerce Security**; Dominant Publishers
- Bible, Eric Cole, Ronald Krutz, James W. Conley; **Network Security**; Wiley India Pvt. Ltd. (Second Edition)
- E. Maiwald; **Fundamentals of Network Security**; McGraw Hill

* * *

MN–2D (Pr): Cyber Security Lab

1 Credit | 30 Minimum Class Hours | Semester VIII

Objective:

The objective of the course is to enable students to–

- Learn the foundations of cyber security and threat landscape.
- Equip students with the technical knowledge and skills needed to protect and defend against cyber threats.
- Develop skills in students that can help them plan, implement, and monitor cyber security mechanisms to ensure the protection of information technology assets.
- Expose students to responsible use of online social media networks.
- To systematically educate the necessity to understand the impact of cyber crimes and threats with solutions in a global and societal context.

Learning Outcome:

After completion of this course, a student will be able to–

- Understand the cyber security threat landscape.
- Develop a deeper understanding and familiarity with various types of cyber attacks, cyber crimes, vulnerabilities and remedies thereto.
- Analyze and evaluate existing legal framework and laws on cyber security.
- Analyze and evaluate the digital payment system security and remedial measures against digital payment frauds.
- Analyze and evaluate the importance of personal data its privacy and security.
- Analyze and evaluate the security aspects of social media platforms and ethical aspects associated with use of social media.

Experiment List

Unit I: Overview of Cyber Security

1. Checklist for reporting cyber crime at cyber crime Police Station.
2. Checklist for reporting cyber crime online.
3. Demonstration of email phishing attack and preventive measures.

Unit II: Overview of Social Media and Security

1. Basic checklist, privacy, and security settings for any one social media platform (e.g., Instagram, Facebook, X, LinkedIn, YouTube, etc.).
2. Reporting and redressal mechanism for violations and misuse of social media platforms.

Unit III: E–Commerce and Digital Payments

1. Configuring security settings in Mobile Wallets and UPIs.
2. Checklist for secure Net Banking.

Unit IV: Digital Devices Security, Tools and Technologies for Cyber Security

1. Setting and configuring two factor authentication in the Mobile Phone.

2. Managing Application permissions in Mobile Phone.
3. Installation and configuration of computer Anti–virus.
4. Installation and configuration of Computer Host Firewall.
5. Wi–Fi security management in Computer and Mobile Phone.

Note: Additional lab assignments may be included based on topics covered in the theory paper.

* * *