



**KOLHAN UNIVERSITY, CHAIBASA  
JHARKHAND, INDIA**

Syllabus of

**Bachelor of Computer Applications  
(BCA)**

**Under Four Year Under Graduate Programme (FYUGP)**

## Table of Content

<b>Sl. No</b>	<b>Content</b>	<b>Page No</b>
<b>1</b>	<b>Syllabus – Semester I</b>	<b>6 – 14</b>
<b>2</b>	<b>Syllabus – Semester II</b>	<b>15 – 21</b>
<b>3</b>	<b>Syllabus – Semester III</b>	<b>22 – 27</b>
<b>4</b>	<b>Syllabus – Semester IV</b>	<b>28 – 49</b>
<b>5</b>	<b>Syllabus – Semester V</b>	<b>50 – 66</b>
<b>6</b>	<b>Syllabus – Semester VI</b>	<b>67 – 81</b>
<b>7</b>	<b>Syllabus – Semester VII</b>	<b>82 – 88</b>
<b>8</b>	<b>Syllabus – Semester VII</b>	<b>89 – 96</b>

**Course Structure for Bachelor of Computer Applications (BCA) Programme Under  
Four Year Under Graduate Programme (FYUGP)**

Sem.	Paper Code	Paper Title	Credits	L-T-P	Total Credits	Contact Hours
SEM - I	CC-1		6		6	90
	CC-2		2		2	30
	CC-3		2		2	30
	IRC-1	Fundamentals of Internet	3	2-1-0	3	45
	IVS-1A		3		3	45
	MJ-1 (Th)	Computer Fundamentals and Introduction to Programming using C	4	4-0-2	6	60
	MJ-1 (Pr)	Programming using C Lab	2			60
SEM - II	CC-4		6		6	90
	CC-5		2		2	30
	CC-6		2		2	30
	IRC-2		3		3	45
	IVS-1B		3		3	45
	MJ-2 (Th)	Object Oriented Programming using Java	4	4-0-2	6	60
	MJ-2 (Pr)	Object Oriented Programming using Java Lab	2			60
SEM - III	CC-7		3		3	45
	CC-8		3		3	45
	CC-9		3		3	45
	IRC-3		3		3	45
	IAP	Internship/Apprenticeship/Project	4	0-0-4	4	60
	MJ-3 (Th)	Database Management System	4	4-0-2	6	60
	MJ-3 (Pr)	Database Management System (SQL) Lab	2			60
SEM - IV	MJ-4 (Th)	Python Programming Language	4	4-0-2	6	60
	MJ-4 (Pr)	Python Programming Lab	2			60
	MJ-5 (Th)	Data Structure using C	4	4-0-2	6	60
	MJ-5 (Pr)	Data Structure using C Lab	2			60
	MN-1 (Th)	Web Technologies	4	4-0-2	6	60
	MN-1 (Pr)	Web Technologies Lab	2			60
	VS-1	Software Engineering	4	3-1-0	4	60
SEM - V	MJ-6 (Th)	GUI Programming using VB.NET	4	4-0-2	6	60
	MJ-6 (Pr)	VB.NET Programming Lab	2			60
	MJ-7 (Th)	Operating System and Introduction to Linux	4	4-0-2	6	60
	MJ-7 (Pr)	Linux Command and Shell Programming Lab	2			60
	MN-2 (Th)	Introduction to Data Science	4	4-0-2	6	60
	MN-2 (Pr)	Data Science using Python Lab	2			60
	VS-2 (Th)	Numerical Methods using Python	2	2-0-2	4	30
	VS-2 (Pr)	Numerical Methods using Python Lab	2			60
SEM - VI	MJ-8 (Th)	Data Communication and Computer Network	4	4-0-2	6	60
	MJ-8 (Pr)	Data Communication and Computer Network Lab	2			60
	MJ-9 (Th)	Web Development using JSP & MySQL	4	4-0-2	6	60
	MJ-9 (Pr)	Web Development using JSP & MySQL Lab	2			60
	MN-3	Business Intelligence	6	5-1-0	6	90
	VS-3	Digital Logic Design and Computer Organisation	4	3-1-0	4	60

<b>SEM - VII</b>	AMJ-1 (Th)	Data Mining	4	4-0-2	6	60
	AMJ-1 (Pr)	Data Mining Lab	2			60
	AMJ-2 (Th)	Big Data Analytics	4	4-0-2	6	60
	AMJ-2 (Pr)	Big Data Analytics Lab	2			60
	RC-1	Research Methodology	6		6	90
	RC-2	Research Proposal	4		4	60
<b>SEM - VIII</b>	AMJ-3 (Th)	Artificial Intelligence	4	4-0-2	6	60
	AMJ-3 (Pr)	Artificial Intelligence Lab	2			60
	AMJ-4 (Th)	Information Security	4	4-0-2	6	60
	AMJ-4 (Pr)	Information Security Lab	2			60
	RC-3	Research Internship/Field Work	4		4	60
	RC-4	Research Report	4		4	60
	VSR	Project Management	2	2-0-0	2	30
<b>Total Credits</b>						<b>176</b>

**Exit Options:**

1. After Semester – II: Exit option with **Certificate in Computer Application** (with a minimum of 44 credits).
2. After Semester – IV: Exit option with **Diploma in Computer Application** (with a minimum of 88 credits).
3. After Semester – VI: Exit option with **Bachelor of Computer Application** (with a minimum of 132 credits).
4. After Semester – VIII: Award of **Bachelor of Computer Application (Honours/Research)** (with total of 176 credits).

**Prepared by,**

**1. Mr. Riki Sutradhar**

Assistant Professor (Contractual),  
Department of Vocational Courses (B.C.A.),  
Jamshedpur Co-operative College, Jamshedpur

**2. Mr. Subodh Kumar**

Assistant Professor (Contractual),  
Department of Vocational Courses (B.C.A.),  
Jamshedpur Co-operative College, Jamshedpur

# **PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES**

## **Bachelor of Computer Applications (BCA)**

### **Programme Outcomes**

1. Acquire Knowledge of mathematical foundations, 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.
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.

# IRC–1: Fundamentals of Internet

3 Credits | 45 Minimum Class Hours | Semester I

**Objective:** The objective of the course is to–

- Define internet.
- Explain various terminology used in internet.
- Explain the application of internet.
- Explain various threats, cybercrimes, and law.

## Learning Outcomes:

After completion of this course, a student will have an understanding of what the internet is, its purpose, function for users, and various cyber threats.

## Outline of the Course

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

Unit	Topic	Minimum Class Hours
I	Introduction to Internet & Internet Technology	7
II	Internet Terminologies	10
III	Internet Threats	14
IV	Cybercrime and Cyber Law	14
<b>Total</b>		<b>45</b>

## Detailed Syllabus

### Unit I: Introduction to Internet and Internet Technology

Introduction to Internet, History of Internet, Advantages of Internet, Applications of Internet for Business Development.

Network and Its Types and Topologies, Internet Connections, Dial-Up Connection, Direct Connection and Broad Band Connection, Internet Address, URL, ISP, Intranet, Extranet, VPN.

### Unit II: Internet Terminologies

Piconet, Infrared, Bluetooth, 2G and 3G Technology, Modem, Gateways, Routers, Bridge, Hub, Switch, Wi-Fi, Wi-Max, Nano Technology, Web Site, Web Portal, Web Browser, and Web Server.

### **Unit III: Internet Threats**

Identity Protection while using Internet, Information Tracking–IP Address, Cookies, Tracking Scripts, HTTP Referrer, User Agent, Private Browsing and Anonymous Browsing, Confidentiality of Information–Practices to ensure Confidentiality of Information, Computer Security: Introduction, Threats to Computer Security – Viruses (Worms, Trojans), Spyware, Adware, Spamming, PC Intrusion, Phishing, Eavesdropping, Solutions to Computer Security Threats, Firewall.

### **Unit IV: Cybercrime and Cyber Law**

Introduction, Common Forms of Cybercrime – Cyber Trolls and Bullying, Cyber Stalking, Spreading Rumours Online, Online Fraud, Information Theft–Phishing, Online Scams, Illegal Downloads, Child Pornography, Reporting Cybercrime, Computer Forensics – Important Practices of Computer Forensics, Cyber Law and IT Act – Introduction, India’s IT Act and IT (Amendment) Act, 2008.

#### **Recommended Books:**

- **The Internet Book** by Douglas E. Comer (PHI Publication)

#### **Further readings:**

- **Internet And Introduction**, CI Stens School of Computing (TMH Publication)

# MJ–1 (Th): Computer Fundamentals and Introduction to Programming using C

4 Credits | 60 Minimum Class Hours | Semester I

## Objective:

The objective of the course is to introduce the fundamentals computer system and C programming language.

## 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		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Fundamentals of Computer	3
II	Data Representation	3
III	C Fundamentals	10
IV	Control Structures and C Preprocessor	12
V	Arrays, Strings, Pointers, and Functions	20
VI	Structures, Unions and File Handling	12
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Fundamentals of Computer

**Structure and Working of Computer:** Functional block diagram of computer (Central Processing Unit, Arithmetic Logic Unit, Control Unit, Input/Output Unit, Memory Unit, Disk Storage Unit, Bus Structure).

**Input/Output Devices:** Input Devices (Keyboard, Mouse, Scanner, MICR, OMR), Output Devices (VDU, Printers, Plotter, Projector, Speakers).

**Computer Memory:** Primary memory (RAM, ROM, Sequential access memory, Cache memory, Virtual memory), Secondary memory (Magnetic tape, Magnetic disk, Optical disk, Floppy disk, External hard drive, Solid state drive, USB flash drive).



**Programming Language:** Low level language (Machine language, Assembly language), High level language (Procedural-oriented language, Problem-oriented language, Natural language), Pseudo code, Flowchart.

**Computer Software:** System software, Utility software, Application software.

## Unit II: Data Representation

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

**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 III: C Fundamentals

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 {scanf( ), printf( ), getch( ), getche( ), getchar( ), gets( ), putchar( ), putchar( ), puts( )}, Character functions {isalpha( ), isdigit( ), isalnum( ), isspace( ), islower( ), isupper( ), isxdigit( ), iscntrl( ), isprint( ), ispunct( ), isgraph( ), isblank( ), tolower( ), toupper( )}.

## Unit IV: Control Structures and C Preprocessor

**Control Structures:** Sequence structure, Selection structure (if statement, if-else statement, if-else if-else statement, switch-case statement), Loop structure (while, do-while, for loop), Other statements (break, continue, goto, exit).

**C Preprocessor:** Types of C preprocessor directives, Comparison of macros with functions, File Inclusion.

## Unit V: Arrays, Strings, Pointers, and Functions

**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 {strlen( ), strlwr( ),strupr( ), strcat( ), strncat( ), strcpy( ), strncpy( ), strcmp( ), strncmp( ), strcmpi( ), stricmp( ), strnicmp( ), strdup( ), strchr( ), strrchr( ), strstr( ), strset( ), strnset( ), strrev( )}.

**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), Recursion.

## Unit VI: Structures, Unions and File Handling

**Structures:** Definition, Declaration, Initialization, Accessing structure elements, Array of structures, Pointers and structures, Passing structures to a function.

**Union:** Definition, Declaration, Initialization, Accessing structure elements, Differentiate between structure and union, Enumerated data type.

**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, Standard function {fopen( ), fclose( ), feof( ), fseek( ), rewind( )}, Using text files {fgetc( ), fputc( ), fprintf( ), fscanf( ), etc.}.

### Recommended Books:

- Floyd, T. L., **Digital Fundamentals** (Fifth Edition), New Delhi: Pearson Education, 2002
- Hamacher, V. C.; Z. G. Vranesic; S. G. Zaky, **Computer Organization** (Fourth Edition), New Delhi: Tata McGraw-Hill, 1996
- 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): Programming using C Lab

2 Credits | 60 Minimum Class Hours | Semester I

## Objective:

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, Unions and File Handling to solve the real world problems.
- Illustrate memory allocation to variables dynamically and perform operations on text and binary files.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

## Experiment List

### Unit I: C Fundamentals

1. Write a program to evaluate the arithmetic expression  $((A + B / B * D - E) * (F - G))$ . Read the values 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 perform addition of two numbers without using '+' operator.
4. Write a program to find ASCII value of a character entered by user.
5. Write a program to find quotient and remainder by a division process.
6. Write a program to find the size of int, float, double and char data type.
7. Write a program to swap two numbers without using temporary variable.

## Unit II: Control Structures and C Preprocessor

1. Write a program to find the largest number among three numbers.
2. Write a program to find all roots of a quadratic equation.
3. 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).
4. Write a program to find factorial of a given number.
5. Write a program to generate Fibonacci series up to  $N^{\text{th}}$  term.
6. Write a program to check whether a number is palindrome or not.
7. Write a program to check whether a number is prime or not.
8. Write a program to check whether a N digits number is Armstrong number or not.
9. Define a macro with one parameter to compute the volume of a sphere. Write a program using this macro to compute the volume for spheres of radius 5, 10 and 15 meters.
10. 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.
11. Write symbolic constants for the binary arithmetic operators +, -, \*, and /. Write a program to illustrate the use of these symbolic constants.

## Unit III: Arrays, Strings Pointers, and Functions

1. Write a program to find the sum of all elements, average of all elements, and the second largest integer in a “One–Dimensional” integer array.
2. Write a program to swap first and last element of a “One–Dimensional” integer array.
3. 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).
4. Write a program to reverse a string. Switch case string
5. Write a program to count number of vowels, consonants and spaces in a given string.
6. 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).

7. Write a program to concatenate two strings using pointer.
8. Write a program to find the length of a string using pointer.
9. 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()” function) and deallocate (i.e., “free()” function) array memory.
10. Write a program to read an integer array and find maximum and minimum number in the array. The program should input total number of elements (limit) and elements in array from user. Use dynamic memory allocation to allocate (i.e., “calloc()” function) and deallocate (i.e., “free()” function) array memory.
11. 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., “calloc()” function) and deallocate (i.e., “free()” function) array memory. After that use “realloc()” function to alter the size of existing allocated memory blocks for the integer array and print the array.
12. Write program that use function to return the greatest common divisor of two given integers.
13. Write a program to print the transpose of a given matrix using function.
14. Write a program that calculates the nth Fibonacci series using recursive function.

#### **Unit IV: Structures, Unions and File Handling**

1. 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.
2. Create a union named ‘**Book**’ containing ‘**book\_id**’, ‘**title**’, ‘**author\_name**’ and ‘**price**’. Write a program to pass the union as a function argument and print the book details.
3. Write a program to display the contents of a file.

4. Write a program to copy the contents of one file to another file.
5. Write a program to create a text file named “**MyInfo.txt**”, open it, type-in some information about yourself and count the number of characters in the file.

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

## MJ–2 (Th): Object Oriented Programming using Java

4 Credits | 60 Minimum Class Hours | Semester II

### Objective:

The course will introduce students to object oriented programming using Java. It assumes that students know the basics of scalar types (integers, strings and booleans) and fundamental control structures in procedural programming (loops, assignment statements, conditional expressions). It will focus on more sophisticated features such as design of classes, inheritance, interfaces, packages and APIs. It will also cover the basic principles of event handling, multithreading, exception handling, swing programming and JDBC.

The main objectives of the course are as follows:

- Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods etc.
- Understand the fundamentals of object-oriented programming in Java, including defining classes, objects, invoking methods etc. and exception handling mechanisms.
- Understand the principles of inheritance, packages and interfaces.
- Gain knowledge in the concepts of exception handling, swing and JDBC.

### Learning Outcomes:

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

- Analyze the logic of a given problem.
- Use branching control statements and iterative control statements.
- Achieve multiple inheritance using interface.
- Understand Java Swing and JDBC to design application.

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Introduction and Fundamentals of Java	12
II	Classes, Objects and Methods	12
III	Arrays, Strings, and Wrapper classes	8
IV	Packages, Multithreading and Exception Handling	8
V	Managing Input/ Output Files in Java, Java Collection Framework	8
VI	Event and GUI programming, JDBC	12
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Introduction and Fundamentals of Java

Concepts of Object Oriented Programming, Benefits of OOP, History and Features of Java, C/C++ vs Java, Java Runtime Environment, Java Virtual Machine, Java Development Kit, Java Compiler And Interpreter, Java Program Structure ,Implementing a Java Program, Tokens- Keywords, Identifiers, Constants, Data Types- Primitive & Non- Primitive Data Types, Object Reference Types, Strings, Variable declaration & initialization, Types of variables such as local, instance, and static variables, Java Operators and Expressions, Type Conversion in Expressions, Operator Precedence and Associativity, Mathematical Functions, Input/ Output in Java, Command Line Arguments.

**Control Structures:** Decision Making and Branching Statements - if Statement, The if.. else Statement, Nesting of if ... else Statements, The else if Ladder, The switch Statement, Iteration or Loop Statements - while Loop, do ... while Loop, for Loop, for each Loop, Labeled Loop, Nested Loops, Jump Statements - break and continue Statements, return Statement.

### Unit II: Classes, Objects and Methods

Defining a Class, Fields declaration, Method declaration, Creating object, Anonymous object in Java, Accessing class members, Access or Visibility Modifier, this keyword, Method overloading, Constructors, Constructor overloading, static members.

**Inheritance and Polymorphism:** Inheritance Basics, Super and Sub class, Types of Inheritance, Overriding methods, super keyword, final (variables, methods and classes), Static and Dynamic Binding, Abstract methods and classes.

**Interfaces:** Defining Interfaces, Implementing Interfaces, Extending Interfaces, Accessing Interface variable, Multiple Inheritance through Interfaces, Classes V/s Interfaces.

### Unit III: Arrays, Strings and Wrapper classes

One Dimensional Arrays, Declaration, Creation, Initialization of Arrays, Two Dimensional Arrays, String class and its methods, Wrapper Classes.

### Unit IV: Packages, Multithreading and Exception Handling

**Package:** Introduction, Java API Packages, Using System Packages, Naming Conventions, Creating Packages, Accessing Packages, Using a Package, Adding a Class to a Package, Hiding Classes.

**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.

### **Unit V: Managing Input/ Output Files in Java, Java Collection Framework**

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 VI: Event and GUI programming, JDBC**

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

**Swings:** Architecture, Components of Swing- JLabel, JButton, JCheckBox, JRadioButton, JList, JComboBox, JTextField, JTextArea, JPanel, JFrame etc., Working With Graphics, Working with Colour, Adding And Removing Controls, Responding To Controls, Layout Managers (Flow Layout, Grid Layout, Card Layout, Border Layout), Handling Events.

**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-2 (Pr): Object Oriented Programming using Java Lab

2 Credits | 60 Minimum Class Hours | Semester II

## Objective:

The main objectives of the course are as follows:

- To teach the students basics of JAVA programs and its execution.
- To make the students learn concepts like packages and interfaces.
- To teach applications using Console I/O and File I/O, GUI and JDBC applications.

## 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.
- Understand and Apply Object oriented features and Java concepts.
- Apply the concept of multithreading and implement exception handling.
- Access data from a Database with java program.
- Develop applications using Console I/O and File I/O, GUI applications.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

## 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 ( $x^y$ ), log and square root depending upon the user input.
3.	Write a program to find roots of a quadratic equation, $ax^2 + bx + c = 0$ . Implement following conditions – <ul style="list-style-type: none"><li>• If the discriminant is positive, then display two distinct real roots.</li><li>• If the discriminant is zero, then display two equal roots.</li><li>• If the discriminant is negative, then display two distinct complex roots.</li></ul>
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 list of numbers reading input as command line argument.

6.	Write a menu based program to check Prime, Armstrong, Automorphic and Krishnamurty number.
7.	Write a menu based program to convert following – <ul style="list-style-type: none"> <li>• Decimal to Binary Number</li> <li>• Binary to Decimal Number</li> </ul>

### Unit II: Arrays, Strings and Wrapper Classes

1.	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.
2.	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. <i>(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.)</i>
3.	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
4.	Write a program to count number the of palindrome words in a sentence and display the longest palindrome word.
5.	Write a program to implement different methods of wrapper class.
6.	Write a program to convert primitive to wrapper class and wrapper to primitive.

### Unit III: Classes, Objects and Methods

1.	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”.
2.	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.
3.	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.

4.	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.
5.	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.
7.	Write a program to demonstrate the multiple inheritance using interfaces.

#### **Unit IV: Packages, Multithreading and Exception Handling**

1.	Write a program to make a package Balance in which has Account class with Display_Balance method in it. Import Balance package in another program to access Display_Balance method of Account class.
2.	Write a java program to implement thread life cycle.
3.	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.
4.	Write a Java program to use the try, catch and finally block to handle the exception.
5.	Write a program to illustrate the throws keyword in Java.
6.	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 V: Managing Input/ Output Files in Java, Java Collection Framework**

1.	Write a java program to copy the contents from one file to other file.
2.	Write a java program to read the student data from user and store it in the binary file.
3.	Write a program to add, retrieve and remove the element from the ArrayList.
4.	Write a program to implement LinkedList and perform different operations on it.
5.	Write a program to implement push() and pop() operation on Stack in Java using java Collection.

#### **Unit VI: Event and GUI programming, JDBC**

1.	Java program to create a simple calculator with basic operations such as +, -, /, * using java swing elements. Use appropriate layout.
----	--

2.	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.
3.	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-3 (Th): Database Management System

4 Credits | 60 Minimum Contact Hour | Semester III

## Objective:

The objective of the course is to provide course is to present 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 analyze database requirements and determine the entities involved in the system and their relationship to one another.
- To know about the manipulate a database using SQL
- Understand Normalization techniques, Transaction processing, Concurrency Control techniques and Recovery of databases against crashes are also covered.

**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		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Introduction to Databases	12
II	Data Modeling Using Entity-Relationship(ER) Model	12
III	Relational Model and Relational Algebra	16
IV	Structured Query Language (SQL) and PL/SQL	12
V	Transaction Management and Concurrency Control, Distributed Database Management System	08
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Introduction to Databases

*Databases and Database Users:* Introduction, An example, Characteristics of the Database Approach, Advantages of Using DBMS Approach, A Brief History of Database Applications, When Not to Use a DBMS, Components of Database.

*Database System Concepts and Architecture:* Data Models, Schemas, and Instances, Three-schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client-Server Architectures, Classification of Database Management Systems.

### Unit II: Data Modeling Using Entity-Relationship(ER) Model

Using High-Level Conceptual Data Models for Database Design, A Sample Database Application, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Refining the ER Design for the Sample Database, ER Diagrams, Naming Conventions and Design Issues, Codd's Rules, Conceptual Object Modeling using UML Class Diagrams.

### Unit III: Relational Model and Relational Algebra

*Relational Model:* Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions and Dealing with Constraint Violations, Unary Relational Operations: SELECT and PROJECT, Relational Algebra Operations from SET Theory, Binary Relational Operations: JOIN and DIVISION, Additional Relational Operations, Examples of Queries in Relational Algebra.

*Relational Database Design:* Anomalies in a database, Functional Dependency, Normal Forms (1NF, 2NF, 3NF, BCNF), Lossless join and Dependency, Normalization through Synthesis, Higher Order Normal Forms.

### Unit IV: Structured Query Language (SQL) and PL/SQL

*Introduction to various Databases:* Oracle-SQL, MySQL, SQL Server, DB2, MS Access.

*Structured Query Language (SQL):* Introduction, SQL Environment, Classification of SQL Statements, The ISO SQL Data Types, SQL Operators, Integrity Constraints, Data Definition—Creating a Database, Creating a Table, Changing a Table Definition, Removing a Table, Creating an Index, Removing an Index, Views, Granting and Revoking Privileges to Users.

*Data Manipulation:* Inserting, Updating & Deleting Data from database, Simple Queries, More Complex SQL Queries, Aggregate Functions, Order by Clause, Group by Clause, Having Clause, Joins, Sub Queries, Correlated Sub Queries.

***PL/SQL***: Introduction to PL/SQL, Advantages of PL/SQL, The Generic PL/SQL Block, The PL/SQL Execution Environment, Identifiers and Keywords, Operators, Declarations, Assignments, Control Statements, Exceptions, Cursors, Subprograms, Triggers , Stored Procedures, Functions, and Packages.

## **Unit V: Transaction Management and Concurrency Control, Distributed Database Management System**

***Transaction Management and Concurrency Control***: What is transaction, Properties of Transactions, Database, Concurrency control, Serializability and Recoverability, Concurrency control with locking Methods, Concurrency control with time stamping methods, concurrency control with optimistic methods, Database Recovery, Database Security.

***Distributed Database Management System***: Evolution of Distributed Database Management System (DDBMS), DDBMS Advantages and Disadvantages, Characteristics of DDBMS, DDBMS Components, and Distributed Concurrency Control.

### **Recommended Books:**

- Abraham Silberschatz, Henry F. Korth, S. Sudharshan, ***“Database System Concepts”***, 7<sup>th</sup> 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-3 (Pr): Database Management System (SQL) Lab

2 Credits | 60 Minimum Contact Hours | Semester III

**Objective:** The main objectives of the course are as follows:

- To know about the analyze database requirements and determine the entities involved in the system and their relationship to one another.
- To know about the manipulate a database using SQL commands and PL/SQL programs.

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

- Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data.
- Gain knowledge on how to use SQL for Creating, Modifying and Accessing tables in Database.
- Implement Order by and Group by clauses.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

## 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) Remove the table from the database.
- (i) Change the name of the table from Employee to KU\_Emp

**Unit II: UPDATE and DELETE Statements**

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)

- 2.
- (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**

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

- 3.
- (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.

**Unit IV: ROLLBACK and COMMIT Statements**

Create Teacher table with the following fields(Name, DeptNo, Date\_of\_joining, DeptName, Location, Salary)

- 4.
- (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

**Unit V: SELECT Statement**

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)

- 5.
- (a) Create above tables with appropriate data types
  - (b) Insert details of three departments and details of 5 employees.
  - (c) List the employee 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.

6.	<p>Consider the table Dept (deptno, dname, loc) Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno) created earlier, write following query:</p> <ul style="list-style-type: none"> <li>(a) To list all employee who do not have commission.</li> <li>(b) To list all 'Salesman' of dept. 30 who have commission.</li> <li>(c) To list employee name and annual salary (sal*12).</li> <li>(d) To list the name of all employees whose name begins with letter 'Raj'.</li> <li>(e) Display empno, ename, deptno and dname of all employee.</li> <li>(f) List the name of those employees who earn more than all of the employees of 'Sales' dept.</li> </ul>
<b>Unit VI: ORDER BY and GROUP BY Clause, Aggregate Functions</b>	
7.	<p>Create a table Library with appropriate data type of following structure: (Book_id, Title, Author, Subject, Publisher, Quantity, Price)</p> <ul style="list-style-type: none"> <li>(a) Insert any ten records</li> <li>(b) Calculate total quantity of books of each subject.</li> <li>(c) Calculate average price of books of each publisher.</li> <li>(d) Display total quantity, maximum and minimum price of subjects 'Java' and 'Python'.</li> <li>(e) Display list of all the books with price more than 300 in ascending order of price.</li> <li>(f) List Author wise count of books in alphabetical order.</li> </ul>
<b>Unit VII: Join and Sub-Query</b>	
8.	<p>Consider the table Dept (deptno, dname, loc) Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno) created earlier, write following query:</p> <ul style="list-style-type: none"> <li>(a) Display the manager who is having maximum number of employees working under him?</li> <li>(b) List the names of employees, who take highest salary in their departments.</li> <li>(c) Create a view Emp_Dept , which contains Employee name, job, salary and department name.</li> </ul>
<b>Unit VIII: GRANT and REVOKE Statement</b>	
9.	<ul style="list-style-type: none"> <li>(a) Create a new user 'ku' having password 'ku123'</li> <li>(b) Grant all privileges to the user 'ku'</li> <li>(c) Create a table BCA(adm_id, name)</li> <li>(d) Revoke all privileges from the user 'ku'</li> <li>(e) Grant only CREATE and SELECT privilege from user 'ku' on table BCA</li> <li>(f) Revoke SELECT privilege from user 'ku' from table BCA.</li> </ul>
<b>Unit IX: PL/SQL Programming</b>	
10.	<ul style="list-style-type: none"> <li>(a) Write a PL/SQL program to demonstrate Cursors.</li> <li>(b) Write a PL/SQL program to demonstrate Functions.</li> <li>(c) Write PL/SQL queries to create Procedures.</li> <li>(d) Write PL/SQL queries to create Triggers.</li> </ul>

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

# MJ-4 (Th): Python Programming Language

4 Credits | 60 Minimum Class Hours | Semester IV

**Objective:** 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. This course is also to help the students to get familiar with: The main objectives of the course are as follows:

- To Introduce Python Programming Language as Multipurpose Programming Language with Features and Applications.
- To Learn Installing Python and Introducing Cross Multiplatform Usage of Python.
- To Practice Basic Language Features of Python and Implement Oops Concepts Using Python.
- Learn core python structures and flow control, Create and run python functions
- Explore the python library functions for various purpose
- Object Oriented Programming using Python
- Learn Files Handling in Python
- Learn GUI Programming and Databases operations 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.
- Build package and modules in Python with reusability and exception aspect.
- Explain how to design GUI Applications in Python and evaluate different database operations.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Introduction and Python Fundamentals	12
II	Flow of Control	8
III	Sequences-String, List, Tuples and Dictionaries	10
IV	Functions, Python Libraries and File Handling	10
V	Object Oriented Programming using Python	8
VI	Interface Python with MySQL and Python GUI Programming	12
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Introduction and Python Fundamentals

**Introduction to Python Programming:** History, Features, Application, Working in Default CPython Distribution-Interactive and Script Mode (Python IDLE), Working with Jupyter Notebook and Sympder IDE, Running Python program, Python Character Set, Tokens-Keywods, 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 math module (pi, e, sqrt, ceil, floor, pow, fabs, sin, cos, tan); random module (random, randint, randrange), statistics module (mean, median, mode).

### Unit II: 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 III: Sequences-String, List, Tuples and Dictionaries

**String:** Traversal, operations – concatenation, repetition, membership; functions/methods–len(), capitalize(), title(), upper(), lower(), count(), find(), index(), isalnum(), islower(), isupper(), isspace(), isalpha(), isdigit(), split(), partition(), strip(), lstrip(), rstrip(), replace() etc.; String slicing.

**Lists:** Definition, Creation of a list, Traversal of a list. Operations on a list - concatenation, repetition, membership; functions/methods–len(), list(),append(), extend(), insert(), count(), index(), remove(), pop(), reverse(), sort(), min(), max(), sum(); Lists Slicing; Nested lists; finding the maximum, minimum, mean of numeric values stored in a list; linear search on list of numbers and counting the frequency of elements in a list.

**Tuples:** Definition, Creation of a Tuple, Traversal of a tuple. Operations on a tuple - concatenation, repetition, membership; functions/methods – len(), tuple(), count(), index(), sorted(), min(), max(), sum(); Nested tuple; Tuple slicing; finding the minimum, maximum, mean of values stored in a tuple; linear search on a tuple of numbers, counting the frequency of elements in a tuple.

**Dictionary:** Definition, Creation, Accessing elements of a dictionary, add an item, modify an item in a dictionary; Traversal, functions/methods – len(), dict(), keys(), values(), items(), get(), update(), del(), del, clear(), fromkeys(), copy(), pop(), popitem(), setdefault(), max(), min(), count(), sorted() copy() etc.;

#### **Unit IV: Functions, Python Libraries and File Handling**

**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, Creating and Using Libraries/ Packages, Installing Packages with *pip* installer. Using NumPy, SciPy and Matplotlib library.

**File Handling:** Need for a data file, Types of file: Text files, Binary files and CSV (Comma separated values) files. Basic operations on a text file: Open (filename – absolute or relative path, mode) / Close a text file, Reading and Manipulation of data from a text file, Appending data into a text file, standard input / output and error streams, relative and absolute paths. Binary File: Basic operations on a binary file: Open (filename – absolute or relative path, mode) / Close a binary file, Pickle Module – methods load and dump; Read, Write/Create, Search, Append and Update operations in a binary file. CSV File: Import csv module, functions – Open / Close a csv file, Read from a csv file and Write into a csv file using csv.reader ( ) and csv.writerow( ).

**Exceptions:** Built-in Exceptions, Handling Exceptions, Exception with Arguments, User defined Exceptions.

#### **Unit V: Object Oriented Programming using Python**

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

#### **Unit VI: Interface Python with MySQL and Python GUI Programming**

Connecting to MySQL from Python, Creating Cursor Instance, Executing SQL Query, Extracting Data from Resultset, Parameterised Query, Insert, Update and Delete Operations.

**Python GUI Programming:** Introduction, Creating GUI Applications using tkinter library, Adding Widgets.

#### **Recommended Books:**

- Introduction to Problem Solving with Python - E. Balagurusamy, TMH
- Problem Solving and Programming with Python, Reema Thareja, Oxford University Press

#### **Further Readings:**

- Introduction to computation and Programming using Python ,JohnV.Guttag, (PHI Learning)

- An Introduction to Computer Science using Python 3 – Jason Montojo, Jennifer Campbell, Paul Gries, 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-4 (Pr): Python Programming Lab

2 Credits | 60 Minimum Class Hour | Semester IV

### Objectives:

- 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.

**Outcomes:** Upon completion of the course, students 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.

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

### Experiment List

Unit I: Data Types, Operator & Expression and Input/ Output	
1.	Write a program to demonstrate different data types 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 input a 6 digit number and divide it into three 2 digit numbers from left side. Print all number with their sum For example – <i>Input: 475963</i> <i>Output: 47 59 63</i> <i>Sum: 169</i>
4.	Write a program to calculate amount payable after compound interest.
5.	Write a program to generate 6 digit random secure OTP between 10000 to 99999.
6.	Write a program to swap two numbers without any arithmetic operator or temporary variable.
7.	Write a program find sum, difference and product of two complex numbers.
8.	Write a program to show the difference between equals to (==) and identity (is) operator.



## Unit II: Flow of Control (if-else, loop)

1.	Write a python program to input three numbers and display the sum and difference of largest and smallest number.									
2.	<p>Input three numbers and calculate two sums as per following :</p> <p>Sum1: as the sum of all input numbers Sum2: as the sum of non-duplicate numbers and one of the duplicate numbers. Sum will be 0 if all are same.</p> <p>For example –</p> <table style="width: 100%; border: none;"> <tr> <td style="text-align: center;">Input: 5 2 5</td> <td style="text-align: center;">Input: 5 2 4</td> <td style="text-align: center;">Input: 5 5 5</td> </tr> <tr> <td style="text-align: center;">Sum1 = 12 (5+2+5)</td> <td style="text-align: center;">Sum1 = 11 (5+2+4)</td> <td style="text-align: center;">Sum1 = 15 (5+5+5)</td> </tr> <tr> <td style="text-align: center;">Sum2 = 7 (5+2)</td> <td style="text-align: center;">Sum2 = 11 (5+2+4)</td> <td style="text-align: center;">Sum2 = 0</td> </tr> </table>	Input: 5 2 5	Input: 5 2 4	Input: 5 5 5	Sum1 = 12 (5+2+5)	Sum1 = 11 (5+2+4)	Sum1 = 15 (5+5+5)	Sum2 = 7 (5+2)	Sum2 = 11 (5+2+4)	Sum2 = 0
Input: 5 2 5	Input: 5 2 4	Input: 5 5 5								
Sum1 = 12 (5+2+5)	Sum1 = 11 (5+2+4)	Sum1 = 15 (5+5+5)								
Sum2 = 7 (5+2)	Sum2 = 11 (5+2+4)	Sum2 = 0								
3.	Write a program to find the roots of quadratic equation. Implement all conditions.									
4.	Write a program to input some numbers repeatedly and print their sum. The program ends when users say “no more to enter” or program aborts when the number entered is a negative number. <i>Do not use list, tuple etc.</i>									
5.	<p>Write a python script to display first N Fibonacci series numbers starting from 0 and 1. Print ‘<b>p</b>’ as suffix to those numbers which are also prime.</p> <p>For example –</p> <p style="text-align: center;">For N=10 Fibonacci series : 0, 1, 1, 2p, 3p, 5p, 8, 13p, 21, 34</p>									
6.	<p>Write a program to find sum of the series without using math module or exponent (**) operator:</p> $1+x-x^2/2! + x^3/3! - x^4/4! + \dots\dots\dots x^n/n!$									
7.	<p>Print following pattern using loop-</p> <pre style="margin-left: 40px;"> 1 121 12321 1234321 123454321 </pre>									

## Unit III: String, Tuple, List and Dictionary

1.	Write a python program to check whether the given string is palindrome or not.
2.	<p>Write a program that should do the following –</p> <ul style="list-style-type: none"> <li>• ask the user for a string</li> <li>• extract all the digits from the string</li> <li>• If there are digits: <ul style="list-style-type: none"> <li>✓ sum the collected digits together</li> <li>✓ print out the original string, the digits and the sum</li> </ul> </li> <li>• If there are no digits: print the original digit and a message “has no digits”</li> </ul>
3.	<p>Write a program to input a line of text and create a new line of text where each word is reversed.</p> <p>For example: Input String: I love Python.</p>

	<i>Output String: Ievol nohtyP.</i>
4.	Write a program to input a line of text and count following – Alphabets, Uppercase and Lowercase alphabets, Vowels, Consonants, Digits, Words and Special characters
5.	Write a program to compare two strings and the display the characters of largest string in pyramid pattern. For example: <i>String 1: Chaibasa</i> <i>String 2: Jamshedpur</i> <i>Largest String is Jamshedpur</i> J Ja Jam Jams Jamsh Jamshe Jamshed Jamshedp Jamshedpu Jamsnedpur
6.	Create a list and perform the following methods (a) insert() (b) remove() (c) append() (d) len() (e) pop() (f) clear()
7.	Write a program to input a list of integers, replicates it twice and then print the sorted list in ascending and descending order. For example- <i>Original list: [12, 45, 77]</i> <i>Replicated list: [12,45,77,12,45,77]</i> <i>Sorted in ascending order : [12,12,45,45,77,77]</i> <i>Sorted in descending order: [77,77,45,45,12,12]</i>
8.	Write a program to count the frequency of each numbers of a list.
9.	Write a program to move all duplicate values in a list to the end of the list.
10.	Write a program to create a nested / two dimensional (2D) list of integers and display all elements.
11.	Write a program to input n numbers into a tuple and calculate mean, median and mode of tuple's element.
12.	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.
<b>Unit IV: Functions, Python Libraries, File Handling and Exceptions</b>	
1.	Write a function that takes a number <i>n</i> and then returns a randomly generated number having exactly <i>n</i> digits e.g., if <i>n</i> is 2 then function can randomly returns a number between 10 and 99.

2.	Write a Python function that takes two lists and returns True if they are equal otherwise false.
3.	Write a function <i>convertNum (dec, base)</i> 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”.
4.	Write a python program to find factorial of a number using recursion.
5.	Demonstrate a python code to print try, except and finally block statements.
6.	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 and modules from MyMath package into a program and perform basic operations.
7.	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.
8.	Python program to read a file and capitalize the first letter of every word in the file.
9.	Write a function remove_lowercase() that accepts a filenames and copy all lines that starts with capital letter to "capital.txt" and small letter to "small.txt".
10.	Write a program to create a binary file “Book.dat” having structure [BookNo, Book_Name, Author, Price]. <ul style="list-style-type: none"> <li>• Write a user defined function CreateFile() to input data for a record and add to Book.dat .</li> <li>• Write a function ShowBooks() to display details of all books.</li> <li>• Write a function SearchBook(BookNo) to search a book.</li> <li>• Write a function ModifyBooks(BookNo, Price) which accept bookno and price in argument and increment the price of that book.</li> <li>• Write a function CountRec(Author) in Python which accepts the Author name as parameter and count and return number of books by the given Author stored in the binary file “Book.dat”</li> </ul>
11.	Write a program to create a CSV file “user.csv” which will contain user name and password for some entries. Read the “user.csv” file and display all records.
<b>Unit V: Object Oriented Programming using Python</b>	
1.	Write a Python class to convert an integer to a roman numeral
2.	Python program to create a class in which one method accepts a string from the user and another prints it.
3.	Python program to create a class which performs basic calculator operations.
4.	Write a python program to create <i>Account</i> class with <i>deposit, withdraw</i> function and then implement it. Use constructor to initialize <i>initial balance</i> . Inherit a class <i>Customer</i> from <i>Account</i> having fields account number and customer name. Override a method <i>display()</i> of Account class into Customer to display customer details with balance amount.

5.	<p>Create a Student class and initialize it with <i>name</i>, <i>roll</i>, and <i>age</i>. Make methods to :</p> <p>(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.</p>
<b>Unit VI: Interface Python with MySQL and Python GUI Programming</b>	
1.	<p>Write a complete application on database operations on student data such as [Roll, Name, Marks]. The program should perform following task –</p> <p>(a) Connection to the database  (b) Creating the student table  (c) Saving / Inserting records  (d) Reading records  (e) Updating records  (f) Deleting records</p>
2.	<p>Design a GUI application using tkinter module having two text boxes which will accept two numbers from user and display their sum in another text box.</p>

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

## MJ-5 (Th): Data Structure using C

4 Credits | 60 Minimum Class Hours | Semester IV

**Objective:** Data Structure is considered as one of the fundamental paper towards a more comprehensive understanding of programming and application development. Student is expected to work towards a sound theoretical understanding of Data Structures and also compliment the same with hands on implementing experience.

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.
- Able to walkthrough algorithm.

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Introduction to Data Structure	8
II	Arrays	12
III	Stacks & Queues	14
IV	Linked list	14
V	Trees	12
<b>Total</b>		<b>60</b>

### Detailed Syllabus

#### Unit I: Introduction to Data Structure

Definition, Types of data structures - Primitive & Non-primitive, Linear and Non-linear, Operations on data structures.

Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - malloc, calloc, realloc and free. Algorithm Specification, Performance Analysis, Performance Measurement.

## Unit II: Arrays

**Basic Concepts:** Definition, Declaration, Initialisation, Operations on arrays, Types of array-Single Dimensional Array, Representation of Linear Arrays in memory, Multidimensional arrays, Representation of multidimensional arrays.

**Sorting:** Introduction, Selection sort, Bubble sort, Quick sort, Insertion sort, Merge Sort, Performance Analysis

**Searching:** Linear Search, Binary Search – Iterative and Recursive method, Performance Analysis

## Unit III: Stacks & Queues

**Stacks:** Basic Concepts – Definition and Representation of stacks; Array Implementation of Stack, Operations on 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; Application of queue.

## Unit IV: Linked list

Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly linked list, Circular linked list; Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation, Linked list implementation of Stacks and Queues (Dynamically Linked Stacks and Queues)

## Unit V: Trees

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, Expression Tree.

### 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-5 (Lab): Data Structure using C Lab

2 Credits | 60 Minimum Class Hour | Semester IV

### Objectives:

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

### Learning Outcomes:

- 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.

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

### Experiment List

Unit I: Array, Searching and Sorting	
1.	Write a program to insert a new element to a given position in an array.
2.	Write a program to delete an element from an existing array from the given position.
3.	Write a program to merge two strings into a third string.
4.	Write a function to search an element in the array using linear search.
5.	Write a function to search an element in the array using binary search.
6.	Write a program to define a function to sort an array using bubble sort in ascending/descending order.
7.	Write a program to define a function to sort an array using selection sort in ascending/descending order.
8.	Write a program to define a function to sort an array using insertion sort in ascending/descending order.
9.	Write a program to define a function to sort an array using merge sort in ascending/descending order.
10.	Write a program to define a function to sort an array using quick sort in ascending/descending order.

11.	Write a program to define functions add(A,B) and multiply(A,B) to add and multiply two matrices A and B passed to the parameter.
12.	Write a program to define a function which returns transpose of a matrix.
<b>Unit II: Stack</b>	
1.	Write a menu based program to implement push, pop and traversal operations on a stack using array.
2.	Write a menu based program to implement push, pop and traversal operations on a dynamic stack using linked list.
3.	Write a program to convert an infix expression to postfix using stack operations.
<b>Unit III: Queue</b>	
1.	Write a menu based program to implement insert, delete and traversal operations on a queue using array.
2.	Write a menu based program to implement insert, delete and traversal operations on a dynamic queue using linked list.
3.	Write a program that implements Circular Queue using arrays.
<b>Unit IV: Linked List</b>	
1.	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.
<b>Unit V: Tree</b>	
1.	Write a program to create a binary search tree and perform Insertion and different types of traversal.

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



# MN-1 (Th): Web Technologies

4 Credits | 60 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, scripting languages, and CSS. Dynamic web page development is also being discussed to make the student competent to design dynamic 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 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		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Introduction to Web Technology	10
II	Introduction to Markup Languages	15
III	Web Page Designing With CSS	10
IV	Scripting Language, JavaScript and DOM	15
V	DHTML	10
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Introduction to Web Technology

**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

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: Web Page Designing With CSS**

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 IV: Scripting Language, JavaScript and DOM**

**Scripting Language:** Introduction to Scripting Language (Client-Side Scripting Language, Server-Side Scripting Language), Applications of Scripting Language, Advantages and Disadvantages of Scripting Language, Differences between Scripting Language and Compiled Language.

**JavaScript:** 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}.

**DOM:** Introduction to DOM (Core DOM, XML DOM, HTML DOM), Tree representation of HTML Document, innerHTML, JavaScript HTML DOM Elements (Finding HTML elements by id, Finding HTML elements by tag name, Finding HTML elements by class name, Finding HTML elements by CSS selectors, Finding HTML elements by HTML object collections).

### **Unit V: DHTML**

**DHTML:** Introduction to DHTML, Components of DHTML, Uses of DHTML, Features of DHTML, Differences between HTML and DHTML, DHTML JavaScript, JavaScript and HTML event, JavaScript and HTML DOM, CSS with JavaScript in DHTML, DHTML CSS, DHTML Events (onabort, onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onload, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onreset, onselect, onsubmit, onunload).

**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
- Michele Petrovsky; **Dynamic HTML in Action**; Tata McGraw Hill Publications

**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

# MN–1 (Pr): Web Technologies Lab

2 Credits | 60 Minimum Class Hours | Semester IV

## Objective:

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

## Learning Outcome:

After completion of this course, a student will be able to design and implement 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		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

## 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, cellpadding, cellspacing, background, width, and height) wherever necessary. A specimen of the Time–Table is given below–

### 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	L U N C H	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–

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

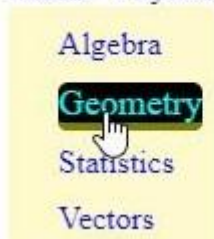
6. Create a HTML page that displays a form with the information – First Name, Last Name, Gender, Subject of Interest, Language Known, Date of Birth (DD-MM-YYYY), Contact No., Address, Pin Code, State, and Country with Reset and Submit button in tabular format.

### Unit III: 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 IV: Scripting Language, JavaScript and DOM

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 of **Question No. 1 in Unit: III** 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.
6. Create a HTML page that contains a simple table with 4 rows and 3 columns. Write a JavaScript function to add rows to the table.

#### Unit V: DHTML

1. Create a HTML page that contains a sentence. Write a JavaScript function to change the font color of the sentence using DHTML event “onclick”.
2. Students should develop a **Mini Project** using the concept learnt in this course (i.e., table, frame, form, CSS, JavaScript, DOM, and DHTML). 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.*

# VS-1: Software Engineering

4 Credits | 60 Minimum Contact Hours | Semester IV

## Objective:

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.
- Understand the UML notation.
- 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		Total Marks
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	100 + N/A = 100

Unit	Topic	Minimum Class Hours
I	Software Engineering Concepts, Process Model and Project Management	15
II	Requirements Analysis and Specification	15
III	Software Design	10
IV	Coding and Testing of Software	10
V	Software Quality Management and Maintenance	10
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Software Engineering Concepts, Process Model and Project Management

**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, Scheduling- PERT Charts, Gantt Charts.

### Unit II: Requirements Analysis and Specification

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.

**Object Oriented Modeling:** UML Introduction, Use Case Diagrams, Class Diagrams

### Unit III: Software Design

Classification, Software Design Approaches, Cohesion and Coupling, Function Oriented Software Design, Introduction to Object Oriented Design.

**User Interface Design:** Need of UI design, Design issues, The UI design Process, User analysis, User Interface Prototyping, Interface Evaluation.

### Unit IV: Coding and Testing of Software

**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 Quality Management and Maintenance

**Software Quality Management:** ISO 9000 and SEI/CMM and their Comparison.

**Software Maintenance:** 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

# MJ-6 (Th): GUI Programming using VB.NET

4 Credits | 60 Minimum Contact Hours | Semester V

**Objective:** VB.NET is a simple, multi-paradigm object-oriented programming language designed to create a wide range of Windows, Web, and mobile applications built on the .NET Framework

The main objectives of the course are as follows:

- To gain expertise in Visual Basic Programming, Application Creation, Variables and Calculations, Decision Making, Lists and Loops, Procedures and Functions, Multiple Forms, Modules, and Menus, Arrays, Files, Printing, Structures, and Database Integrations
- Students will be exposed to the following concepts and/or skills at an Intermediate concepts level: Analyze program requirements, Design/develop programs with GUI interfaces, Code programs and develop interface using Visual Basic.Net

**Learning Outcomes:** After completing this course-

- Use Visual Studio IDE to design application.
- Develop GUI Application using Form Controls and its events.
- Apply Object Oriented concepts in GUI Application.
- Use Data access controls to store data in Database and retrieve it.
- Use Data Binding in GUI Application

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Introduction to .Net Technology	4
II	The Visual Basic.Net Language	8
III	Control statements and Arrays	10
IV	OOP using VB.Net	12
V	Windows Form	14
VI	Data access with ADO.Net & Report	12
Total		60

## Detailed Syllabus

### Unit I: Introduction to .Net Technology

Introduction to .Net, The .Net Framework Class Library, Working with the .Net FCL, Namespaces, Types of a .Net Namespace.

## **Unit II: The Visual Basic.Net Language**

VB.NET Features, Advantages of VB.NET, Working with Visual Studio IDE, Creating Applications with Visual Basic.NET, Keywords, Literals and Variables, VB.Net Data types, Operators.

## **Unit III: Control Statements and Arrays**

Decision Statements- If..then, If..then..else, Select.. Case, Loop Statements- While, Do ..Loop, For .. Next, For Each ..Next, Arrays.

## **Unit IV: OOP using VB.Net**

Object Oriented features- Abstraction, Encapsulation, Polymorphism, Inheritance, Declaring Classes, Implementing Typecasting, Procedures and Functions, Optional arguments, Error handling in Procedures, Properties, Public and Private variables, Types of Properties, Polymorphism, Inheritance, Method Overriding.

## **Unit V: Windows Form**

Introduction to Class Libraries, Events, Delegates and Event Handlers, Windows Application, Windows GUI, First Win Forms Application, Controls, Text controls, Selection List Controls, VB.Net is overridden, Some controls with examples. Error handling In Windows Forms: Types of Validations, Types of Errors, Exceptions, Classified Runtime based Exceptions. SDI and MDI Applications: SDI and MDI interfaces, Characteristics of MDI components, Creating MDI Forms.

## **Unit VI: Data Access with ADO.Net & Report**

Overview of Microsoft Database Access Technology, ADO.Net, Creating a Database, ADO.Net Architecture, ADO.Net Class Libraries, Databound Controls, Creating Report in .NET Using Crystal Reports.

### **Recommended Books:**

- Jeffrey R. Shapiro –Visual Basic.net The Complete Reference - TMH

### **Further Readings:**

- Dave Grundgeiger, Programming Visual Basic.Net, O'Reilly
- Visual Basic .NET Programming Black Book Holzner Steven Dreamlech Press, 2015, New Delhi

## MJ-6 (Pr): VB.NET Programming Lab

2 Credits | 60 Minimum Contact Hours | Semester V

**Objective:** VB.NET is a simple, multi-paradigm object-oriented programming language designed to create a wide range of Windows, Web, and mobile applications built on the .NET Framework

The main objectives of the course are as follows:

- To gain expertise in Visual Basic Programming, Application Creation, Variables and Calculations, Decision Making, Lists and Loops, Procedures and Functions, Multiple Forms, Modules, and Menus, Arrays, Files, Printing, Structures, and Database Integrations
- Students will be exposed to the following concepts and/or skills at an Intermediate concepts level: Analyze program requirements, Design/develop programs with GUI interfaces, Code programs and develop interface using Visual Basic.Net

**Learning Outcomes:** After completing this course-

- Use Visual Studio IDE to design application.
- Develop GUI Application using Form Controls and its events.
- Apply Object Oriented concepts in GUI Application.
- Use Data access controls to store data in Database and retrieve it.
- Use Data Binding in GUI Application

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

### Experiment List

1. Calculate area of circle using console application.
2. Calculate area of triangle using windows application.
3. Currency conversion program using windows application.
4. Salary calculation program using windows application.
5. Program to retrieve images using Combo box and Button.
6. Program to simulate a Paint Brush program.
7. Program to move an object using Timer control
8. Program to develop a math calculator.
9. Develop an application which makes use of String and String Builder class.
10. Create a Text Pad application
11. Develop an application using menu.
12. Create an application using Common Dialog Controls.
13. Create an application on database using ADO.NET with a Windows Forms to create, read, update, and delete records in Access/ SQL Server databases.
14. Program on Reports Generations: Visual Studio Reports, Crystal Reports, Deploying Applications.

**Note:** Additional lab assignments may be included as per topics covered in the theory paper.

# MJ-7 (Th): Operating System and Introduction to Linux

4 Credits | 60 Minimum Class Hours | Semester V

## Objective:

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 processes, memory management, file management, Input/Output management and the potential problem of deadlocks. The students will also learn about the Linux operating system.

## Learning Outcome:

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 and memory management.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Overview	10
II	Process Management	15
III	Process Synchronization	07
IV	Memory Management	15
V	Storage Management	07
VI	Linux Operating System	06
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Overview

**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

**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 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

**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).

### Unit IV: Memory Management

**Memory Management Strategies:** Introduction to Memory Management, 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)}]

### Unit V: Storage Management

**File System:** 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 VI: Linux Operating System

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-7 (Pr): Linux Command and Shell Programming Lab

2 Credits | 60 Minimum Class Hours | Semester V

## Objective:

The main objective of this course is to make students familiar with the Linux command-line environment and develop the skills of shell scripting. This course serves as a platform for the subsequent labs related to Process Management, Process Scheduling, etc.

## Learning Outcome:

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.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

## Experiment List

### Unit I: Basic Linux Commands

1. 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.
2. 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\_7* using **mkdir** command.
  - (iv) Change the current directory to *MJ\_7* using **cd** command.
  - (v) Create a text file *MJ\_7\_Info.txt* using **vi** command (or, any text editor) and write basic information about your seventh major course (i.e., Th: Operating System and Introduction to Linux and Pr: Linux Command and Shell Programming Lab). Save and close it.
  - (vi) Copy the contents of *MJ\_7\_Info.txt* to *MJ\_7\_Info\_Copy.txt* using **cp** command.
  - (vii) Open *MJ\_7\_Info\_Copy.txt* using **vi** command (or, any text editor) and verify that it is a copy of *MJ\_7\_Info.txt*. Close the file.
  - (viii) Rename *MJ\_7\_Info\_Copy.txt* as *Major\_7\_Info.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** command and delete the whole directory *BCA\_Copy* using **rm -r** command.
3. List all the files contained in your home directory including hidden files.
  4. Display the sorted list of all files contained in your home directory by file size.
  5. 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*.
  6. 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*.
  7. Write a **sed** command that swaps the first and second words in each line in a file.
  8. 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*.
  9. Download the file [https://www.nta.ac.in/Download/FAQ\\_TPC\\_STUDENTS.pdf](https://www.nta.ac.in/Download/FAQ_TPC_STUDENTS.pdf) from the web to your current directory.
  10. 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: Different Commands for Process Management in Linux

1. Launch a process (or, program or, application) in the foreground from terminal. 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 using **jobs** command (**jobs** command is only available in the *cs*, *bash*, *tcsh*, and *ksh* shells).
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.
7. Display a dynamic real-time table of processes of your Linux operating system using **top** command.
8. Display the snapshot of currently-running processes using **ps** command.



9. How do we terminate a running process on our Linux operating system?
10. Report the used and available space in the primary memory storage in human-friendly units like megabytes or kilobytes.
11. Report the used and available space in the secondary memory storage in human-friendly units like megabytes or kilobytes.
12. How can you prioritize a process as per your requirement?

### **Unit III: Programming In Shell Script**

1. Write a program in Shell Script to see current date, time, username, and current directory.
2. Write a program in Shell Script to evaluate the arithmetic expression  $((A + B / B * D - E) * (F - G))$ . Read the values A, B, C, D, E, F, and G from the standard input device.
3. 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.
4. The distance between Ayodhya and Dwarka (i.e., 1724 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.
5. The length and breadth of a rectangle and radius of a circle are input through the keyboard. Write a program in Shell Script to calculate the area and perimeter of the rectangle as well as the area and circumference of the circle.
6. An integer is input through the keyboard. Write a program in Shell Script to find out whether it is an odd or even number.
7. Write a program in Shell Script which receives any year from the keyboard and determines whether the year is leap or not. If no argument is supplied, the current year should be assumed.
8. In a company, different employees are paid as follows: If the basic salary of an employee is less than Rs. 5000, then HRA = 10% of basic salary and DA = 90% of basic salary. If the basic salary of an employee is either equal to or above Rs. 5000, then HRA = Rs. 900 and DA = 98% of basic salary. If the employee's salary is input through the keyboard, write a program in Shell Script to find the gross salary of employees.
9. If the cost price and selling price of an item is input through the keyboard, write a program in Shell Script to determine whether the seller has made profit or incurred loss. Also determine how much profit was made or loss incurred.
10. If a five digit number is input through the keyboard, write a program in Shell Script to calculate the sum of its digits.
11. If a five digit number is input through the keyboard, write a program in Shell Script to reverse its digits.
12. Write a program in Shell Script to find the factorial of any number entered through the keyboard.
13. Write a program in Shell Script to generate Fibonacci series up to  $N^{\text{th}}$  term.

14. Write a program in Shell Script to check whether a number is Armstrong number or not.
15. Write a program in Shell Script to check whether a number is Palindrome number or not.
16. Write a program in Shell Script to check whether a number is Prime number or not.
17. The marks obtained by a student in five different subjects are input through the keyboard. The student gets a division as per the following rules:
  - Percentage above or equal to 60 – First division
  - Percentage between 50 and 59 – Second division
  - Percentage between 40 and 49 – Third division
  - Percentage less than 40 – FailWrite a program in Shell Script to find the division obtained by the student.
18. Write a program in Shell Script to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs. 12 per hour for every hour worked above 40 hours. Assume that employees do not work for fractional part of an hour.
19. Two numbers are entered through the keyboard. Write a program in Shell Script to find the value of one number raised to the power of another.
20. Write a program in Shell Script to store integer elements into a “one–dimensional” array and display the elements on the screen.
21. Write a program in Shell Script to find element using Linear Search.
22. Write a program in Shell Script to convert a decimal number to its binary equivalent.
23. Write a program in Shell Script to store integer elements into a “two–dimensional” array and display the elements on the screen.
24. Write a program in Shell Script to store integer elements into a “two–dimensional” array and add the elements of each row of the array.
25. Write a program in Shell Script to store integer elements into a “two–dimensional” array and find largest element in the array.
26. Write a program in Shell Script to store integer elements into a “two–dimensional” array and find the transpose of the array.
27. Write a program in Shell Script to store integer elements into two “two–dimensional” array and add the two arrays.
28. Write a program in Shell Script to store integer elements into two “two–dimensional” array and multiply the two arrays.
29. 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.
30. 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.

**31.** 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.*

# MN–2 (Th): Introduction to Data Science

4 Credits | 60 Minimum Contact Hours | Semester V

## Objective:

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 completing this course-

- 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 the 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		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

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
IV	Computers in Data Science	15
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Data Science - An Overview

Introduction to Data Science, Definition and description of Data Science, history and development of 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, users of Data Science and its hierarchy, overview of different Data Science techniques, challenges and opportunities in business analytics, different industrial application of Data Science techniques.

### **Unit II: Mathematics and Statistics in Data Science**

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**

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.

### **Unit IV: Computers in Data Science**

Role of computer science in Data Science, various components of computer science being used for Data Science, role of relation data base systems in Data Science: SQL, NoSQL, role of data warehousing in Data Science, terms related with data warehousing techniques, importance of operating concepts and memory management, various freely available software tools used in Data Science : R, Python, important proprietary software tools, different business intelligence tools and its crucial role in Data Science project presentation.

### **Recommended Books:**

- Data Smart: Using Data Science to Transform Information into Insight 1<sup>st</sup> Edition by John W. Foreman. (2015) Wiley Publication.
- Data Science from Scratch: First Principles with Python 1st Edition by Joel Grus.

### **Further Readings:**

- Data Science For Dummies by Lillian Pierson (2015)

# MN–2 (Pr): Data Science using Python Lab

2 Credits | 60 Minimum Contact Hours | Semester V

## Objectives:

- 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:** At the end of this course, the students 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. Lab Manual
- Perform correlation and regression analytics on standard data sets
- Present and interpret data using visualization packages in Python.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

## 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 the diabetes data set from 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.
5. Apply and explore various plotting functions on UCI data sets.
  - Normal curves
  - Density and contour plots
  - Correlation and scatter plots
  - Three dimensional plotting

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

## VS–2 (Th): Numerical Methods using Python

2 Credits | 30 Minimum Contact Hours | Semester V

### 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.

The main objectives of the course are as follows:

- This will help you to choose, develop and apply the appropriate numerical techniques for your problem.
- The goal is to provide a basic understanding of the derivation, analysis, and use of these numerical methods.

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

- Perform interpolation, approximation, and integration of Functions
- Solve initial values problems governed by ordinary differential equations

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
30	60	3	3	2	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Errors in Numerical Calculations	2
II	Solution of Algebraic and Transcendental Equations	5
III	Interpolation	6
IV	Numerical Differentiation and Integration	6
V	Numerical Solution of Ordinary Differential Equations	8
VI	Numerical Solution of Linear System of Equations	3
<b>Total</b>		<b>30</b>

### Detailed Syllabus

#### Unit I: Errors in Numerical Calculations

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

## **Unit II: Solution of Algebraic and Transcendental Equations**

Introduction, Bisection Method, Newton-Raphson Method, Method of False Position and Secant Method.

## **Unit III: Interpolation**

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

## **Unit IV: Numerical Differentiation and Integration**

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**

Solution by Taylor's Series, Euler's method

## **Unit VI: Numerical Solution of Linear System of Equations**

Direct Methods- Matrix Inversion Method, Gauss-Jordan Method, Gaussian Elimination Method.

### **Recommended Books:**

- S.S. Sastry , 'Introductory methods of Numerical Analysis' , 5<sup>th</sup> 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



# VS-2 (Pr): Numerical Methods using Python Lab

2 Credits | 60 Minimum Contact Hours | Semester V

## Objectives:

- This will help you to choose, develop and apply the appropriate numerical techniques for your problem using Python.
- To improve the student's skills in numerical methods by using the numerical analysis software and computer facilities.

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

- Solve Non Linear and Linear equations through Python programs.
- Interpolate using various formula through Python
- Find derivatives, integration and solve differential equation using python.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
30	60	3	3	2	2	15	N/A	60	25	75	25	75 + 25 = 100

## Experiment List

Unit I: Errors in Numerical Calculations	
1.	Program to find Absolute, Relative and Percentage error.
Unit II: Solution of Algebraic and Transcendental Equations	
1.	Program for finding roots of Non Linear equation $f(x)=0$ using Bisection method
2.	Program for finding roots of Non Linear equation $f(x)=0$ using Newton Raphson method
3.	Program for finding roots of Non Linear equation $f(x)=0$ using Secant Method
Unit III: Interpolation	
1.	Program to interpolate using Newton Forward Interpolation formula.
2.	Program to interpolate using Newton Backward Interpolation formula.
Unit IV: Numerical Differentiation and Integration	
1.	Program to find value of derivative of a polynomial.

2.	Program for solving Numerical Integration by Trapezoidal Rule.
3.	Program for solving Numerical Integration by Simpson's 1/3 rule.
<b>Unit V: Numerical Solution of Ordinary Differential Equations</b>	
1.	Program to solve ordinary differential equation using Taylor series.
<b>Unit VI: Numerical Solution of Linear System of Equations</b>	
2.	Program to solve system of linear equations using Matrix Inversion method.

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

# MJ–8 (Th): Data Communication and Computer Network

4 Credits | 60 Minimum Class Hours | Semester VI

## Objective:

The objective of the course is to introduce the concepts of computer networks. This course will help students to learn the fundamental layered structure of networks, understand common offered layered services, and examine protocols and algorithms 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.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Overview	10
II	Physical Layer and Media	10
III	Data Link Layer	10
IV	Network Layer	10
V	Transport Layer	10
VI	Application Layer	10
Total		60

## Detailed Syllabus

### Unit I: Overview

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

**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

**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–N Automatic 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)}.

### Unit IV: Network Layer

**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 V: Transport Layer

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 VI: Application Layer

**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); Prentice Hall of India, 1993
- Fred Halsall; **Data Communication, Computer Networks and Opens Systems** (Fourth Edition); Pearson Education, 2000

# MJ-8 (Pr): Data Communication and Computer Network Lab

2 Credits | 60 Minimum Class Hours | Semester VI

## Objective:

The objective of the course is to introduce the concepts of different types of network cables, network devices, and network IP. In this course, students are going to be learn about connecting two personal computers. Students will also be introduced to the network modeling and simulation, and they will have the opportunity to build some simple networking models using Packet Tracer software and perform simulations.

## Learning Outcome:

After completion of this course, a student will be able to evaluate his/her network design approaches and expected network performance.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

## Experiment List

1. Detail study of different types of network cables (Twisted–Pair Cable, Co–Axial Cable, and Fiber–optic Cable) with their connectors.
2. Detail study of different types of network devices (Repeater, Hub, Switch, Modem, Bridge, Router, and Gate Way).
3. Detail study of network IP (Classification of IP address, Sub netting, and Super netting).
4. Connect two personal computers with Windows operating system using a Crossover Wired Cable and transfer some folders or files between the two connected computers.
5. Study of basic network commands (ping, ipconfig, getmac, hostname, nslookup, tracert, netstat, netsh, arp, pathping, and systeminfo) with the help of **Command Prompt** and **Packet Tracer** software.
6. Perform an initial configuration of a Switch using **Packet Tracer** software.
7. Perform an initial configuration of a Router using **Packet Tracer** software.
8. Configure a Network Topology using **Packet Tracer** software

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

# MJ-9 (Th): Web Development using JSP and MySQL

4 Credits | 60 Minimum Class Hours | Semester VI

## Objective:

The objective of the course is to enable students to–

- Understand the concepts underlying technologies in JSP using MySQL.
- Install and configure Java Development Kit (JDK), Web Server (Apache Tomcat Server), and Database (MySQL).
- 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, 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		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Overview of JSP	5
II	Elements, Directives, Implicit Objects, and Standard Action Tags	15
III	Expression Language (EL) and Exception Handling	15
IV	Advanced JSP	15
V	Web Application Development using JSP and MySQL	10
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Overview of JSP

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

**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**

**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)}, Operator Precedence, Reserve words, 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**

**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, Java Database Connectivity with Microsoft Access without DSN, Java Database Connectivity with Microsoft Access with DSN.

**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.

### **Unit V: Web Application Development using JSP and MySQL**

MVC Architecture in JSP, Registration Form, Login Form, File Uploading, File Downloading, Handling Date, Page Redirect, Hits Counter, Auto Refresh, Sending Email.

#### **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-9 (Pr): Web Development using JSP and MySQL Lab

2 Credits | 60 Minimum Class Hours | Semester VI

## Objective:

The objective of the course is to enable students to–

- Understand the concepts underlying technologies in JSP using MySQL.
- Install and configure Java Development Kit (JDK), Web Server (Apache Tomcat Server), and Database (MySQL).
- 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, 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		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

## Experiment List

### Unit I: Overview of JSP

1. Explain step by step procedure to install and Configure of Java Development Kit (JDK).
2. Explain step by step procedure to install and Configure of Web Server (Apache Tomcat Server).
3. Explain step by step procedure to install and Configure of Database (MySQL).

### Unit II: Elements, Directives, Implicit Objects, and Standard Action Tags

1. Create a JSP document to declare and initialize a variable 'count' using Declaration Tag and display it using Scriptlet Tag.
2. Create a JSP document to declare and initialize two variables 'num1' and 'num2'. Add the values of those two variables and display the sum using Scriptlet Tag.
3. Create a JSP document to declare and initialize seven variables 'A', 'B', 'C', 'D', 'E', 'F', and 'G' using Scriptlet Tag. Evaluate the arithmetic expression  $((A + B / B * D - E) * (F - G))$  and display the result using Expression Tag.
4. Create a JSP document to use five attributes associated with Page Directive viz. 'language', 'contentType', 'pageEncoding', 'isELIgnored', and 'import'. Display current date by importing 'Date' class which is from "java.util" package.

5. The “java.math” package has a class called ‘BigInteger’ that lets you create whole numbers with an arbitrary number of digits. Import the class using Page Directive. Create a JSP document to take a large BigInteger number from user, squares it, and display the result.
6. Create a JSP document to detect Language, Country, Local Date, and Formatted Currency. Import necessary Java packages using Page Directive.
7. Create two JSP documents named “header.jsp” and “footer.jsp”. Include those two JSP documents in a main JSP document named “index.jsp” using Include Directive.
8. Make a JSP document named “sign.jsp” with your Name and Email Id as “Signature Block”. Include it in main JSP document named “MyInfo.jsp” using Include Directive.
9. Create a JSP document to display current date and time in AM/PM format using Implicit Object ‘out’.
10. 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’.
11. Create a simple JSP document to redirect the user to the official website of ISRO <https://www.isro.gov.in/> using Implicit Object ‘response’.
12. 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.
13. Create a user defined bean class ‘Calculator’ and inside this class create method to calculate cube of a given number. Create a JSP document to calculate cube of the given number by simply invoking the method of the class ‘Calculator’ using Standard Action Tag “jsp:useBean”.
14. Create a JSP document named “PrintDate.jsp” to display current date. Include this JSP document in a main JSP document named “index.jsp” using Standard Action Tag “jsp:include”.
15. Create a JSP document named “TodaysNews.jsp” that includes a “GoodNews.jsp” document or a “BadNews.jsp” document at random using Standard Action Tag “jsp:include”.

### **Unit III: 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).
3. Create a JSP application that takes the user name from the user and then saves it into the user session. Display the saved data in another page using Implicit Object 'sessionScope' in Expression Language.
4. Create a JSP application that stores the user name in an object of class Cookie. Display the value of the object of class Cookie in another JSP document using Implicit Object 'cookie' in Expression Language.
5. Create a JSP application that sets an attribute (e.g., your name) using application object. Display the attribute in another page using Implicit Object 'applicationScope' in Expression Language.
6. Create a JSP document named "division.jsp" that displays a form with the information – First Number and Second Number with Reset and Submit button in tabular format. After clicking on 'Submit' button, the result of division of those two numbers should display on another JSP document named "process.jsp". Whenever the page "process.jsp" throws an exception, the JSP container should automatically invokes an error page named "error.jsp". To set up the error page "error.jsp", use Implicit Objects 'page' and 'exception'.

#### **Unit IV: Advanced JSP**

1. Create a table 'Student' with field names 'StudentId', 'Name', 'Email', 'ContactNo' in either MySQL or Oracle or Microsoft Access without DSN or Microsoft Access with DSN database. Create a JSP document to fetch all the records from the table 'Employee' and display them.
2. Write a Servlet program to display welcome message in browser using deployment descriptor "web.xml" to map the Servlet.
3. 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.
4. Create a table 'Professor' with field names 'TeacherId', 'Name', 'Email', 'ContactNo' in either MySQL or Oracle or Microsoft Access without DSN or Microsoft Access with DSN 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.
5. Create a JSP application that stores the user name in an object of class Cookie. Display the value of the object of class Cookie in another JSP document.
6. Create a Servlet filter and map in web.xml to logs the IP addresses, current date, and current time of the client from which the requests originate.

## Unit V: Web Application Development using JSP and MySQL

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. Create a JSP document to display current time in AM\PM format by importing 'Date' class which is from "java.util" package and make the JSP auto refresh every second in such a way that the current time changes every second.
3. Create a JSP document to accept a numeric number from the user and display it in English words (For example, 6174 = Six Thousand One Hundred and Seventy Four).
4. 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.*

## MN–3 (Th): Business Intelligence

6 Credits | 90 Minimum Class Hours | Semester VI

### Objective:

The objective of the course is to provide complete knowledge of Business Intelligence. Students will be able to develop logical abilities, which will help them to create programs, applications. The students will also develop an ability, which can easily help them switch to any other language in future.

The main objectives of the course are as follows:

- to provide students with comprehensive and in-depth knowledge of Business Intelligence (BI) principles and techniques by introducing the relationship between managerial and technological perspectives.
- to expose students to the frontiers of BI-intensive BIG data computing and information systems, while providing a sufficiently strong foundation to encourage further research.

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

- Explain the foundations, definitions, and capabilities of DSS, data analytics and BI.
- List the definitions, concepts, and architectures of data warehousing.
- Demonstrate the impact of business reporting, information visualization, and dashboards.
- Outline the definitions, concepts, and enabling technologies of Business Intelligence.

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
90	N/A	3	N/A	6	N/A	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Understanding Business Intelligence	15
II	Mathematical Models for Decision Making	15
III	Classification	15
IV	Business Intelligence Applications	15
<b>Total</b>		<b>60</b>

### Detailed Syllabus

#### Unit I: Understanding Business Intelligence

Effective and timely decisions, Data, information and knowledge, The role of mathematical models, Business intelligence architectures, Ethics and business intelligence, Decision support systems: Definition of system, Representation of the decision-making process, Evolution of information systems, Definition of decision, support system, Development of a decision support system.

## **Unit II: Mathematical Models for Decision Making**

Structure of mathematical models, Development of a model, Classes of models,

Data mining: Definition of data mining, Representation of input data, Data mining process, Analysis methodologies.

Data Preparation: Data validation, Data transformation, Data reduction.

## **Unit III: Classification**

*Classification problems*, Evaluation of classification models, Bayesian methods, Logistic regression, neural networks, Support vector machines,

Clustering: Clustering methods, Partition methods, Hierarchical methods, Evaluation of clustering models.

## **Unit IV: Business Intelligence Applications**

Marketing models: Relational marketing, Sales force management, Logistic and production models: Supply chain optimization, Optimization models for logistics planning, Revenue management systems.

*Data envelopment analysis*: Efficiency measures, Efficient frontier, The CCR model, Identification of good operating practices. Knowledge Management: Introduction to Knowledge Management, Organizational Learning and Transformation, Knowledge Management Activities, Approaches to Knowledge Management, Information Technology (IT) In Knowledge Management, Knowledge Management Systems Implementation, Roles of People in Knowledge Management, Artificial Intelligence and Expert Systems: Concepts and Definitions of Artificial Intelligence, Artificial Intelligence Versus Natural Intelligence, Basic Concepts of Expert Systems, Applications of Expert Systems, Structure of Expert Systems, Knowledge Engineering, Development of Expert Systems.

## **Recommended Books:**

- Business Intelligence: Data Mining and Optimization for Decision Making, Carlo Vercellis, Wiley
- Decision support and Business Intelligence Systems, Efraim Turban, Ramesh Sharda, Dursun Delen, Pearson

## **Further Readings:**

- Fundamental of Business Intelligence, Grossmann W, Rinderle-Ma, Springer

# VS-3: Digital Logic Design and Computer Organization

4 Credits | 60 Minimum Class Hours | Semester VI

## Objective:

The objective of the course is to introduce some of the basic understanding of the design of various functional units and components of a digital computer. The aim is to give an overview of the computer and its functions, with specific references to some of its parts. The student will also get an understanding of the application of Logic Circuit and Boolean algebra in Computer Science and Applications.

## 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.
- 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.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
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	100 + N/A = 100

Unit	Topic	Minimum Class Hours
I	Digital Logic Circuits	12
II	Digital Components	12
III	Register Transfer and Microoperations	12
IV	Basic Computer Organization and Design	12
V	Central Processing Unit	12
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Digital Logic Circuits

**Logic Circuit:** Logic Gates (AND, OR, NOT, NAND, 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, Parallel adder and Serial 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), Timing diagrams.

## Unit II: Digital Components

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)}.

## Unit III: Register Transfer and Microoperations

**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 IV: Basic Computer Organization and Design

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 V: Central Processing Unit

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

# AMJ–1(Th): Data Mining

4 Credits | 60 Minimum Class Hours | Semester VII

**Objective:** This course introduces the fundamental concept of big data analytics. The objective of the course is to provide complete knowledge of big data analytics. Students will be able to develop the concepts of big data analytics with the concept of hadoop framework. The students will also develop an ability which can easily help them to handle big data concept by using the different components of hadoop framework.

The main objectives of the course are as follows:

- Introduce the concepts of big data analytics.
- Introduces the big data framework.
- Students will be familiar with the concept of big data handling concept mechanisms concept with the help of hadoop framework.
- Learn the basic concept of big data analytics process.

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

- Understand the concept & characteristics of big data.
- Explore Hadoop framework and its components. Use HDFS and Map Reduce to analyze . . various industry use cases of big data analytics.
- Learning different Sorting, Shuffling algorithm of mapreduce & understand the concept of big data streaming.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
30	N/A	3	N/A	2	N/A	15	N/A	60	25	75	N/A	75 + N/A = 100

Unit	Topic	Minimum Class Hours
<b>I</b>	Introduction to Data Mining	12
<b>II</b>	Data Processing	10
<b>III</b>	Classification and Prediction	12
<b>IV</b>	Mining Frequent Patterns, Associations, and Correlations	12
<b>V</b>	Cluster analysis	8
<b>VI</b>	Mining Text, and Web Data	6
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Introduction to Data Mining

Data mining Introduction: Definition, Data mining tasks, Data mining as a step of Knowledge discovery process, Applications of Data mining; Data objects and types of attributes,

Recalling mean, median, mode and weighted arithmetic mean.

## **Unit II: Data Processing**

Data quality, Overview of data preprocessing. Classification analysis- definition, Overview of various classification techniques; Decision tree induction- working, examples, specifying attribute test conditions.

## **Unit III: Classification and Prediction**

Evaluating the performance of a classifier- Holdout method, Random sub-sampling, cross validation, Bootstrap; Association analysis: support, confidence, association rules, Frequent Item sets.

## **Unit IV: Mining Frequent Patterns, Associations, and Correlations**

Frequent item set generation - Apriori principle , Apriori algorithm and examples, FP growth algorithm and examples.

## **Unit V: Cluster Analysis**

Introduction, Similarity and Distance Measures, Outliers, Hierarchical Algorithms, Partitional Algorithms (Minimum Spanning Tree, K-Means Clustering, Nearest Neighbor Algorithm).

## **Unit VI: Mining Text, and Web Data**

Text Mining, Mining the World Wide Web

### **Recommended Books:**

- Data Mining: Concepts and Techniques, 3rd edition, Jiawei Han and Micheline Kamber
- Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education.

### **Further readings:**

- Data Mining: A Tutorial Based Primer, Richard Roiger, Michael Geatz, Pearson Education 2003.
- Introduction to Data Mining with Case Studies, G.K. Gupta, PHI 2006
- Insight into Data mining: Theory and Practice, Soman K. P., Diwakar Shyam, Ajay V., PHI 2006
- Data Mining:: Practical Machine Learning Tools and Techniques (Morgan Kaufmann Series in Data Management Systems) by Witten, Frank, Hall

# AMJ–1(Pr): Data Mining Lab

2 Credits | 60 Minimum Class Hours | Semester VII

## Objective:

The main objectives of the course are as follows:

- Students will be familiar with the concept of big data handling concept mechanisms concept with the help of hadoop framework.
- Learn the basic concept of big data analytics process.

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

- Explore Hadoop framework and its components. Use HDFS and Map Reduce to analyze various industry use cases of big data analytics.
- Learning different Sorting, Shuffling algorithm of map reduce & understand the concept of big data streaming.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
30	N/A	3	N/A	2	N/A	15	N/A	60	25	75	N/A	75 + N/A = 100

## Experiment List

1. Listing applications for mining:  
To list all the categorical (or nominal) attributes and the real-valued attributes separately.
2. File format for data mining:  
To study the file formats for the data mining.
3. Conversion of various data files:  
(a) To convert a text file to ARFF(Attribute-Relation File Format) using Weka3.8.2 tool.  
(b) To convert ARFF (Attribute-Relation File Format) into text file.
4. Training the given dataset for an application:  
To apply the concept of Linear Regression for training the given dataset.
5. Testing the given dataset for an application:  
To apply the Navie Bayes Classification for testing the given dataset.
6. Generating accurate models:

To find the good result (by improving the performance) using the training set and testing data set for numerical values.

7. Web mining:

To apply the web mining technique clustering algorithm for the given dataset.

8. Text mining:

To find association between data and to find the frequent item set for text mining.

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

## AMJ–2 (Th): Big Data Analytics

4 Credits | 60 Minimum Contact Hours | Semester VII

**Objective:** This course introduces the fundamental concept of big data analytics. The objective of the course is to provide complete knowledge of big data analytics. Students will be able to develop the concepts of big data analytics with the concept of hadoop framework. The students will also develop an ability which can easily help them to handle big data concept by using the different components of hadoop framework.

The main objectives of the course are as follows:

- Introduce the concepts of big data analytics.
- Introduces the big data framework.
- Students will be familiar with the concept of big data handling concept mechanisms concept with the help of hadoop framework.
- Learn the basic concept of big data analytics process.

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

- Understand the concept & characteristics of big data.
- Explore Hadoop framework and its components. Use HDFS and Map Reduce to analyze . . various industry use cases of big data analytics.
- Learning different Sorting, Shuffling algorithm of mapreduce & understand the concept of big data streaming.

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Introduction to Big Data, Characteristics and its Use Case	8
II	Introduction to Hadoop	12
III	Introduction to YARN and MapReduce	14
IV	Introduction to Big Data Streaming	14
<b>Total</b>		<b>60</b>

### Detailed Syllabus

#### Unit I: Introduction to Big Data, Characteristics and its Use Case

Introduction, Why Big data, What is big Importance of Big Data, Evolution of Big Data, Market Trends, Sources of Data Explosion, Types of Data, Need of Big Data, Big Data and its sources, Characteristics of Big Data, Difference between Traditional IT Approach and Big Data Technology, Capabilities of Big Data, Handling Limitations of Big Data, Technologies Supporting Big Data, Big Data Use Cases.

## **Unit II: Introduction to Hadoop**

Introduction, Why Hadoop, What is Hadoop, History and Milestone of Hadoop, Core Components of Hadoop, Difference between Regular File System and HDFS, Common Hadoop Shell Commands, Hadoop Configuration.

*Hadoop Distributed File System (HDFS)* : Concepts and Architecture, Data Flow (File Read, File Write), Fault Tolerance, Java Base API Different Daemons in Hadoop cluster (NameNode, Secondary NameNode, Job Tracker, Task Tracker and DataNode, Loading a dataset into the HDFS.

## **Unit III: Introduction to YARN and MapReduce**

What is YARN, YARN Infrastructure, Introduction of MapReduce, Analogy of MapReduce, MapReduce Architecture, Example of MapReduce, Sorting, Shuffling, Reducing, Combiner, Creating MapReduce program by using Eclipse.

## **Unit IV: Introduction to Big Data Streaming**

Real time Big Data Streaming, Big data streaming framework, Data streaming process, Tools for big data streaming, Industry use cases for big data streaming

### **Recommended Books:**

- Seema Acharya, Subhashini Chellappan , Big Data and Analytics, Wiley Publication
- Tom White, Hadoop: The Definitive Guide, O'Reilly Media

### **Further readings:**

- Big Data, Black Book: Covers Hadoop 2, MapReduce, Hive, YARN and Data Visualization (2016), DT Editorial Services, Dream tech press

## AMJ–2 (Pr): Big Data Analytics Lab

2 Credits | 60 Minimum Contact Hours | Semester VII

### Objective:

- Students will be familiar with the concept of big data handling concept mechanisms concept with the help of hadoop framework.
- Learn the basic concept of big data analytics process.

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

- Explore Hadoop framework and its components. Use HDFS and Map Reduce to analyze various industry use cases of big data analytics.

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

### Experiment List

1. Install and configure Hadoop, set working directory and various processes.
2. Implement basic commands in Hadoop to manipulate big data:  
  
Ls, Mkdir, Touchz, copyFromLocal or put, cat, copyToLocal or get, moveFromLocal, cp, mv, rmr. du, dus, stat, setrep
3. Create a directory in HDFS at given path.
4. List the contents of a directory.
5. See the contents of a file in Hadoop.
6. Copy a file from source to destination.
7. Copy a file from local file system to HDFS.
8. Display the aggregate length of a file.
9. Write a MapReduce program for wordcount.
10. Implement HDFS and explore its characteristics
11. Explore the properties of YARN.
12. Import data from MySQL into HDFS.
13. Import data from MS Excel into HDFS.
14. Import data from MS Excel into HDFS.
15. Remove a file/directory in HDFS

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



# AMJ-3 (Th): Artificial Intelligence

4 Credits | 60 Minimum Class Hours | Semester VIII

**Objective:** The objective of the course is to introduce the basic concepts of Artificial Intelligence.

**Learning Outcomes:** Upon successful completion of the course, a student will be able to:

- To understand various AI techniques.
- To decide when to use which type of AI technique.

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
30	N/A	3	N/A	2	N/A	15	N/A	60	25	75	N/A	75 + N/A = 100

Unit	Topic	Minimum Class Hours	Marks
I	Introduction to Artificial Intelligence	12	12
II	Introduction to Genetic Algorithms & Knowledge Representation	10	10
III	Forward and backward reasoning & Introduction to Prolog	12	12
IV	Introduction to Natural Language Processing	12	12
V	Reasoning in Uncertain Environments	8	8
<b>Total</b>		<b>60</b>	<b>60</b>

## Detailed Syllabus

### Unit I: Introduction to Artificial Intelligence

Meaning and definition of artificial intelligence, Physical Symbol System Hypothesis, production systems, Characteristics of production systems; Breadth first search and depth first search techniques.

**Heuristic search Techniques:** Hill Climbing, Iterative deepening DFS, bidirectional search. Analysis of search methods. A\* algorithm, and their analysis.

### Unit II: Introduction to Genetic Algorithms & Knowledge Representation

Introduction to Genetic Algorithms Knowledge Representation, Problems in representing knowledge, knowledge representation using propositional and predicate logic, logical consequences, syntax and semantics of an expression, semantic Tableau.

### **Unit III: Forward and backward reasoning & Introduction to Prolog**

*Forward and Backward Reasoning:* Proof methods, substitution and unification, conversion to clausal form, normal forms, resolution, refutation, deduction, theorem proving, in fencing, monotonic and no monotonic reasoning.

*Introduction to Prolog:* Network-based representation and reasoning, Semantic networks, Conceptual Graphs, frames. Description logic (DL), concept language, reasoning using DL. Conceptual dependencies (CD), scripts, reasoning using CD.

### **Unit IV: Introduction to Natural Language Processing**

*Introduction to natural language processing:* Adversarial search and Game theory, Classification of games, game playing strategies, prisoner's Dilemma. Game playing techniques, mini max procedure, alpha-beta cut-offs. Complexity of alpha-beta search. Automated planning, classical planning problem, forward planning, partial order planning, planning with proposal logic, hierarchical task planning, multiagent planning.

### **Unit V: Reasoning in Uncertain Environments**

Reasoning in uncertain environments, Fuzzy logic, fuzzy composition relation, operations on fuzzy sets. Probabilistic reasoning, Bayes theorem, construction of Bayesian networks, belief propagation. Markov processes and Hidden Markov models

#### **Recommended Books:**

- Artificial Intelligent e: Elaine Rich, Kevin Knight, Mc-Graw Hill.
- Introduction to AI & Expert System: Dan W. Patterson, PHI.

#### **Further readings:**

- Artificial Intelligent by Luger (Pearson Education)
- Russel & Norvig, Artificial Intelligent e: A Modern Approach, Pearson Education

## AMJ–3 (Pr): Artificial Intelligence Lab

2 Credits | 60 Minimum Class Hours | Semester VIII

**Objective:** The objective of the course is to introduce the basic concepts of Artificial Intelligence.

**Learning Outcomes:** Upon successful completion of the course, a student will be able to:

- To understand various AI techniques.
- To decide when to use which type of AI technique.

### Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
30	N/A	3	N/A	2	N/A	15	N/A	60	25	75	N/A	75 + N/A = 100

### Experiment List

1. Installation of gnu-prolog, Study of Prolog (gnu-prolog), its facts, and rules.
2. Write simple facts for the statements and querying it.
3. Write a program for Family-tree.
4. Write Program for Monkey-banana Problem.
5. Write a program which behaves a small expert for medical Diagnosis.
6. Write programs for computation of recursive functions, like, factorial, Fibonacci numbers, etc.
7. Write program to solve 5-queens problem.
8. Write a Program for water jug problem.
9. Write a program for travelling salesman program.
10. Case study of standard AI programs, like, Mycin, and AI Shell

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

# AMJ-4 (Th): Information Security

4 Credits | 60 Minimum Class Hours | Semester VIII

## 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.
- Understand the concepts digital signatures, authentication and Blockchain.

## 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		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

Unit	Topic	Minimum Class Hours
I	Introduction to Information Security	09
II	Symmetric Cryptographic Techniques	12
III	Asymmetric Cryptographic Techniques	12
IV	Digital Signatures and Authentication	15
V	Blockchain	12
<b>Total</b>		<b>60</b>

## Detailed Syllabus

### Unit I: Introduction to Information Security

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

### Unit II: Symmetric Cryptographic Techniques

**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**

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**

**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.

### **Unit V: Blockchain**

Introduction, Proof of Work, How Blockchain works?, Types of Blockchain (Public, Private, and Consortium), Need of Blockchain, Cryptocurrency, Blockchain versus Shared Database, Applications of Blockchain Technology, Limitations of Blockchain Technology.

#### **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

## AMJ-4 (Pr): Information Security Lab

2 Credits | 60 Minimum Class Hours | Semester VIII

### 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.
- Understand the concepts digital signatures, authentication and Blockchain.

### 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		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
60	60	3	3	4	2	15	N/A	60	25	75	25	75 + 25 = 100

### 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.*

# VSR: Project Management

2 Credits | 30 Minimum Contact Hours | Semester VIII

**Objective:** The purpose of this course is to prepare the students to understand the tools and techniques involved in a project. Moreover, enable them to execute a project to achieve specific goals that will benefit the organizations and society.

The main objectives of the course are as follows:

- To outline the need for Project Management
- To highlight different techniques of activity planning
- Project Planning & Management

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

- Evaluate and select the most desirable projects.
- Apply appropriate approaches to plan a new project and develop project schedule.
- Identify the important risks facing in a new project

## Outline of the Course

Minimum Class Hours		Exam Time (Hours)		Credits		Marks						
						Semester Internal		End Semester		Full Mark		Total Marks
Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th	Pr	Th + Pr
30	N/A	1½	N/A	2	N/A	15	N/A	35	N/A	50	N/A	50+0 = 50

Unit	Topic	Minimum Class Hours
I	Introduction to Project Management and Project Selection	8
II	Project Planning and Implementation	8
III	Project Monitoring and Control	8
IV	Project Closure	6
<b>Total</b>		<b>30</b>

## Detailed Syllabus

### Unit I: Introduction to Project Management and Project Selection

Objectives of Project Management- Importance of Project Management- Types of Projects Project Management Life Cycle- Project Selection – Feasibility study:  
Types of feasibility Steps in feasibility study.

### Unit II: Project Planning and Implementation

Project Scope- Estimation of Project cost – Cost of Capital – Project Representation

and Preliminary Manipulations - Basic Scheduling Concepts - Resource Levelling – Resource Allocation.

### **Unit III: Project Monitoring and Control**

Setting a base line- Project management Information System – Indices to monitor progress. Importance of Contracts in projects- Teamwork in Project Management - Attributes of a good project team – Formation of effective teams – stages of team formation.

### **Unit IV: Project Closure**

Project evaluation- Project Auditing – Phases of project Audit- Project closure reports Guidelines for closeout reports.

### **Recommended Books:**

- Atkinson, R. (1999). Project management: cost, time and quality.
- Khanna, R. B. (2011), Project Management, PHI Learning Private Limited, New Delhi.

### **Further readings:**

- Berkun, Scott (2008), Making Things Happen: Mastering Project Management, O'Reilly Media: Cambridge, MA.

