

OGY



Walchand Institute of Technology, Solapur  
(An Autonomous Institute)

Affiliated to  
Punyashlok Ahilyadevi Holkar Solapur University,  
Solapur

Choice Based Credit System (CBCS)

Structure and Syllabus  
for  
T. Y. B.Tech. Computer Science and Engineering  
W.E.F. 2025-26

APS

HEAD

Computer Science and Engineering  
Walchand Institute of Technology  
Solapur 413006.

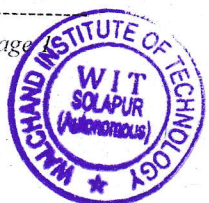
Manish

Dean Academics

Walchand Institute of Technology, Solapur

T.Y.B.Tech.(CSE) Syllabus w.e.f. 2025-26

Page



# Computer Science and Engineering

## Vision

To develop professional engineers in Computer Science & Engineering having ethical values, research aptitude and ability to address challenges of modernization in the IT industry aiming at overall sustainable development of the society.

## Mission

- M1 - To impart quality education in the field of Computer Science & Engineering in accordance with the needs of the Modernization & Globalization through technology enabled education.
- M2 - To inculcate lifelong learning in students to face challenges posed by ever-changing IT career landscape as a disciplined professional with a sense of professional ethics.
- M3 - To inculcate critical thinking and creativity for identifying various societal issues and to provide solutions.
- M4 - To enhance career opportunities for students through academia-industry interaction and research.

# Computer Science and Engineering

## Program Educational Objectives (PEOs)

1. Graduate will exhibit strong fundamental knowledge and technical skills in the field of Computer Science & Engineering to pursue successful professional career, higher studies and research.
2. Graduate will exhibit capabilities to understand and resolve various societal issues through their problem solving skills.
3. Graduate will be sensitive to ethical, societal and environmental issues as a software engineering professional and be committed to life-long learning.

## Knowledge and Attitude Profile (WK)

WK1	A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.
WK2	Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.
WK3	A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.
WK4	Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.
WK5	Knowledge, including efficient resource use, environmental impacts, whole-life cost, re-use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.
WK6	Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.
WK7	Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.
WK8	Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.
WK9	ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

<b>Program Outcomes (POs)</b>	
<b>PO 1</b>	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
<b>PO 2</b>	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
<b>PO 3</b>	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
<b>PO 4</b>	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
<b>PO 5</b>	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
<b>PO 6</b>	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
<b>PO 7</b>	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
<b>PO 8</b>	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
<b>PO 9</b>	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
<b>PO 10</b>	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
<b>PO 11</b>	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
<b>PO 12</b>	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
<b>Program Specific Outcomes (PSOs)</b>	
<ol style="list-style-type: none"> <li>1. Apply the principles of computational mathematics, computer systems and programming paradigms to solve computational problems.</li> <li>2. Design and develop application software with functionalities applicable for desktop, web and mobile applications with due consideration of system software constraints.</li> <li>3. Apply software engineering methods, cutting edge technologie and ICT, using appropriate tools and FOSS alternatives for designing ,developing &amp; testing application software.</li> </ol>	

## Computer Science and Engineering Department

### Legends Used

L	Lecture Hours / week
T	Tutorial Hours / week
P	Practical Hours / week
FA	Formative Assessment
SA	Summative Assessment
ESE	End Semester Examination
ISE	In Semester Evaluation
ICA	Internal Continuous Assessment
POE	Practical and Oral Exam
OE	Oral Exam
MOOC	Massive Open Online Course
HSS	Humanity and Social Science
NPTEL	National Programme on Technology Enhanced Learning
F.Y.	First Year
S.Y.	Second Year
T.Y.	Third Year
B. Tech.	Bachelor of Technology

# Computer Science and Engineering

## Course Code Format

<b>2</b>	<b>1</b>	<b>I</b>	<b>T</b>	<b>U/P</b>	<b>2</b>	<b>C</b>	<b>C</b>	<b>1</b>	<b>T/L</b>
Year of Syllabus revision	Program Code			U-Under Graduate P-Post Graduate	Semester No./ Year1/2/3/...8	CourseType		Course Serial No 1-9	T-Theory, L-Lab session P- Programming

## Program Code

CS	Computer Science and Engineering
----	----------------------------------

## Course Type

BS	Basic Science
----	---------------

ES	Engineering Science
----	---------------------

HU	Humanities & Social Science
----	-----------------------------

MC	Mandatory Course
----	------------------

CC	Core Compulsory Course
----	------------------------

SN*	Self-Learning <i>N* indicates the serial number of electives offered in the respective category</i>
-----	--

EN*	Core Elective <i>N* indicates the serial number of electives offered in the respective category</i>
-----	--

ON*	Open Elective <i>N* indicates the serial number of electives offered in the respective category</i>
-----	--

SK	Skill Based Course
----	--------------------

SM	Seminar
----	---------

MP	Mini project
----	--------------

PR	Project
----	---------

IN	Internship
----	------------

## Sample Course Code

23CSU3CC1T	Discrete Mathematics Structures
------------	---------------------------------

# Computer Science and Engineering

## B. Tech. Semester V

Course Code	Name of Course	Engagement Hours			Credits	FA		SA		Total
		L	T	P		Theory	OE/ POE	ISE	ICA	
23CSU5CC1T	Database Engineering	3	--	--	3	60	--	40	--	150
23CSU5CC2T	Design and Analysis of Algorithm	2	--	--	2	60	--	40	--	125
23CSU5CC3T	Operating Systems	2	--	--	2	60	--	40	--	125
23CSU5EN*4T	Programme Elective Course I	3	--	--	3	60	--	40	--	125
23CSU5CC5P	Advanced Java Programming	1	--	--	1	--	--	25	--	100
23CSU5IK6T	Indian Knowledge System-II	2	--	--	2	--	--	50	--	50
OE03	Open Elective III (Through NPTEL at SY)	2	--	--	2	50	--	--	--	50
MDM03	Multidisciplinary Minor III	3	--	--	3	60	--	40	--	100
<b>Subtotal</b>		18	--	--	18	350	--	275	--	625
<b>Laboratory Courses</b>										
23CSU5CC1L	Database Engineering Lab	--	--	2	1	--	25	--	25	50
23CSU5CC2L	Design and Analysis of Algorithm Lab	--	--	2	1	--	--	--	25	25
23CSU5CC3L	Operating Systems Lab	--	--	2	1	--	--	--	25	25
23CSU5EN*4L	Programme Elective Course I Lab	--	--	2	1	--	--	--	25	25
23CSU5CC5P	Advanced Java Programming Lab	--	--	2	1	--	50	--	25	75
<b>Subtotal</b>		--	--	10	5	--	75	--	125	200
<b>Grand Total</b>		18	--	10	23	350	75	275	125	825

**Note:**

- N\* indicates the serial number of electives offered in the respective category

- ##indicates program code of offering Programme

<b>Computer Science and Engineering</b>										
<b>B. Tech. Semester VI</b>										
Course Code	Name of Course	Engagement Hours			Credits	FA		SA		Total
		L	T	P		Theory	POE	ISE	ICA	
23CSU6CC1T	Artificial Intelligence	3	--	--	3	60		40	--	100
23CSU6CC2T	System Software	3	--	--	3	60		40	--	100
23CSU6CC3T	Software Engineering	3	--	--	3	60		40	--	100
23CSUEN*4T	Programme Elective Course -II	3	--	--	3	60		40	--	100
23CSUEN*5T	Programme Elective Course - III	2	--	--	2			50	--	50
23CSVSEC6P	Full Stack Development	2	--	--	3		50	25	--	75
MDM04	Multidisciplinary Minor IV	2	--	--	3	60		40	--	100
	<b>Subtotal</b>	18	--		18	300	--	275	--	625
<b>Laboratory Courses</b>										
23CSU6CC1L	Artificial Intelligence Lab	--	--	2	1	--		--	25	25
23CSU6CC2L	System Software Lab	--	--	2	1	--		--	25	25
23CSUEN*4L	Programme Elective Course -II Lab	--	--	2	1	--		--	25	25
23CSVSEC6P	Full Stack Development	--	--	2	1	--	50	--	25	25
MDM04	Multidisciplinary Minor IV Lab	--	--	2	1	--		--	25	25
	<b>Subtotal</b>	--	--	10	5	--	50	--	125	125
	<b>Grand Total</b>	18	--	10	23	300	50	275	125	750



**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
**(An Autonomous Institute)**  
**Third Year B.Tech. (Computer Science and Engineering), Semester-V**

**23CSU5CC1T - DATABASE ENGINEERING**

Teaching Scheme		Examination Scheme	
Lectures	3 Hours/week	ESE	60 Marks
Practical	2 Hours/week	ISE	40 Marks
Credits	4	ICA	25 Marks
		OE	25 Marks

**Introduction:**

In today's data-driven economy, no computer science or business curriculum would be complete without a course in databases and data management system. This course emphasizes the understanding of the fundamentals of relational database system including data models, database architectures, normalization, data integrity, security and data manipulation. It also provides students with theoretical knowledge and practical skills in the use of databases and database management systems with the help of Structured Query language (SQL). It ends with covering database transaction and recovery concepts. Upon completion, students should be able to design and implement normalized database structures by creating simple database.

**Course Prerequisite:**

No prerequisite knowledge of databases is required but basic understanding of data-structures and algorithms is assumed. Any general purpose programming language knowledge is needed.

**Course Objectives:**

1. To understand the basics of database design, structure, implementation and applications.
2. To develop the logical design of the database using data modeling concepts such as entity relationship diagrams.
3. To study and use Structured Query Language to query, update, and manage a database.
4. To apply normalization techniques to normalize the database.
5. To familiarize the students with the fundamentals of database transaction processing and learn techniques for concurrency control and recovery methods along with different indexing techniques.

**Course Outcomes:**

At the end of this course, the student will be able to -

1. Apply the principles of database system to design database using data modelling such as E-R and relational model.
2. Formulate the SQL queries to perform CRUD operations.
3. Apply normalization and indexing techniques to optimize the performance of the database.
4. Demonstrate transaction processing techniques and concurrency control protocols.

**Course Instructions:**

Assignments 2 to 4 should be implemented in MySQL/Oracle/ PostGreSQL.  
 Assignments 6 to 11 should be implemented in C++/Java.

<b>Unit – I</b>	<b>Introduction</b>	<b>4 Hours</b>
-----------------	---------------------	----------------

Database System Applications, Purpose of Database Systems, View of data, Database Languages, Data Storage and Querying, Database Architectures, Database users and administrators.		
<b>Unit – II</b>	<b>Database Design and E-R Model</b>	<b>9 Hours</b>
Overview of design process, E-R Model, Constraints, Removing redundant attributes in entity sets, E-R diagrams, Reduction to relational schema, E-R design issues, and Extended E-R features. Relational Model: Structure of relational databases, Database schema, keys, Schema diagrams, Relational Query languages, Relational algebra.		
<b>Unit – III</b>	<b>SQL and Advanced SQL</b>	<b>9 Hours</b>
SQL: Overview, SQL data definition, Basic structure of SQL Queries, Additional basic operations, Set operations, NULL values, Aggregate functions, Nested sub queries, Modification of the databases, Join operations, Views, Transactions, Integrity constraints, SQL data types and schemas, Authorization, Advanced SQL : Embedded SQL, Functions and Procedures, Triggers.		
<b>Unit – IV</b>	<b>Relational Database Design</b>	<b>8 Hours</b>
Features of good Relational Designs, Atomic Domains, First Normal Form, Decomposition using Functional dependencies, Second Normal Form, BCNF, Third Normal Form, Functional-dependency theory.		
<b>Unit – V</b>	<b>Indexing and Hashing</b>	<b>7 Hours</b>
Basic Concepts, Ordered Indices, B+ Tree Index Files, B Tree Index Files, Multiple Key Access, Static Hashing, Dynamic Hashing, Comparison of Indexing and Hashing, Bitmap indices, Index definition in SQL		
<b>Unit – VI</b>	<b>Transactions and Concurrency Control</b>	<b>8 Hours</b>
Introduction, Unsupervised vs. Supervised learning, Applications of Unsupervised learning, Clustering: Clustering as a Machine learning task, different types of clustering techniques, Partitioning methods, k-means algorithm, k-medoids, Hierarchical clustering, Density based methods-DBSCAN, finding patterns using association rule-Definition of common terms, Association rule, Apriori algorithm		
<p><b>Internal Continuous Assessment (ICA):</b>  It should consist of 8-10 laboratory assignments as follows:</p> <ol style="list-style-type: none"> <li>E-R Diagrams: Draw E-R diagram for any specific database application and create a data dictionary for the same.</li> <li> <ol style="list-style-type: none"> <li>Basic SQL DDL commands: write simple queries in SQL on above database application for schema creation and updation.</li> <li>SQL DML commands: insert, update, select command with different clauses, queries using aggregates, grouping and ordering.</li> </ol> </li> <li> <ol style="list-style-type: none"> <li>Nested sub queries, Joins and Set operations: write queries in SQL using concept of nested sub queries, join and different set operations.</li> </ol> </li> <li> <ol style="list-style-type: none"> <li>Views, Integrity constraints and Authorization: queries for creating views, different integrity constraints and authorization commands.</li> <li>Advanced SQL: queries on embedded SQL, functions and procedures, triggers</li> </ol> </li> <li>Convert the created database into 1NF, 2NF, 3NF and BCNF.</li> <li>Given a set of functional dependencies, find canonical cover and closure of functional dependency.</li> <li>Write a Java program for database (created in expt-2) connectivity using JDBC.</li> </ol>		

8. Write a program to implement B+ tree index (n=3 or n=5) on the database previously created.
9. Write a program to implement dynamic hashing on the database previously created.
10. Write a program to simulate any one concurrency control protocol.

#### **Text Books**

1. Database system concepts by Abraham Silberschatz, Henry F. Korth, S. Sudarshan (McGraw Hill International Edition) sixth edition.

#### **Reference Books**

1. Fundamentals of Database systems by Ramez ElMasri, S. B. Navathe (Pearson Education) Fifth edition.
2. Database Management Systems by Ramkrishnan Gehreke (Tata McGraw Hill) third edition.
3. Principles of Database Systems by J. D. Ullman (Galgotia Publications)
4. SQL The Complete Reference, 3rd Edition by James R Groff, Paul N. Weinberg and Andy Opperl
5. Database system concepts by Peter Rob, Carlos Coronel (Cengage Learning) ninth edition.



**23CSU5CC2T - DESIGN AND ANALYSIS OF ALGORITHMS**

Teaching Scheme		Examination Scheme	
Lectures	2 Hours/week	ESE	60 Marks
Practical	2 Hours/week	ISE	40 Marks
Credits	3	ICA	25 Marks

**Introduction:**

This course introduces the algorithms, strategies of algorithm and analysis of algorithm which will help to compare and determine good algorithm.

**Course Prerequisite:**

Student should have knowledge of basic programming. They should also have basic knowledge of data structure and graph theory.

**Course Objectives:**

1. Analyze the asymptotic performance of an algorithm.
2. Apply important algorithmic design paradigms and methods of analysis.
3. Apply algorithm design paradigm to solve real world problem.
4. Identify P, NP, NP-complete and NP-hard problems and differentiate between tractable and intractable problems.

**Course Outcomes:**

- At the end of the course, students will be able to -
1. Derive time and space complexity of a given algorithm.
  2. Select appropriate algorithm design paradigm for a problem.
  3. Apply suitable algorithm design paradigm for a problem.
  4. Describe and distinguish complexity classes of problems.

<b>Unit – I</b>	<b>Introduction</b>	<b>4 Hours</b>
Algorithm Specification: pseudocode conventions, recursive algorithm, Performance Analysis: space complexity, time complexity, asymptotic notations, performance measurement.		
<b>Unit – II</b>	<b>Divide and Conquer</b>	<b>5 Hours</b>
The general method, binary search, finding the maximum and minimum, quick sort, selection sort, merge sort.		
<b>Unit – III</b>	<b>The Greedy Method</b>	<b>6 Hours</b>
The general method, knapsack problem, job sequencing with deadlines, minimum–cost spanning trees – Prim’s and Kruskal’s algorithms, optimal storage on tapes, optimal merge patterns, single source shortest paths.		
<b>Unit – IV</b>	<b>Dynamic Programming</b>	<b>6 Hours</b>
The general method, multistage graphs, all pairs shortest paths, optimal binary search trees, 0/1 knapsack, reliability design, traveling salesperson problem, flow shop scheduling.		

<b>Unit – V</b>	<b>Backtracking</b>	<b>5 Hours</b>
The general method, 8-queens problem, sum of subsets, Hamilton cycle, graph coloring, knapsack problem.		
<b>Unit – VI</b>	<b>NP-Hard and NP-Complete problems</b>	<b>4 Hours</b>
Tractable and Intractable problems: Computability, Halting problem, Computability classes - P, NP, NP-Complete and NP-Hard, Standard NP-Complete problems, NP-Hard problems (only basic problems).		
<p><b>Internal Continuous Assessment (ICA):</b> ICA shall consist of minimum ten practical assignment problems. The nature of the problems shall be with the objective to assess the student’s ability to:</p> <ol style="list-style-type: none"> <li>Compare and choose an appropriate algorithm design paradigm on time and space complexity.</li> <li>Apply algorithm design paradigm to provide a solution to the problem using any programming language the student is proficient in.</li> <li>Effectively assess performance of provided solutions w. r. t. programming language’s runtime implementation.</li> </ol>		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>Fundamentals of Computer Algorithms, Horowitz, Sahni and Rajasekaran (Galgotia Publication).</li> <li>Fundamentals of Algorithms, Gilles Brassard, Paul Bratley (Pearson Publication).</li> <li>Introduction to Algorithms, Thomas Cormen (Pearson Publication).</li> </ol>		
<b>Reference Books</b>		
<ol style="list-style-type: none"> <li>Introduction to Design and Analysis of Algorithms, Goodman (McGraw-Hill Publication).</li> <li>Design and Analysis of Algorithms, Aho, Hopcroft and Ullman (Addison-Wesley Publication).</li> <li>Design and Analysis of Algorithms, Gajendra Sharma (Khanna Publishing House, New Delhi).</li> <li>Design and Analysis of Algorithms, S. Sridhar (Oxford University Press)</li> </ol>		



**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
**(An Autonomous Institute)**  
**Third Year B.Tech. (Computer Science and Engineering), Semester-V**

**23CSU5CC3T - OPERATING SYSTEMS**

Teaching Scheme		Examination Scheme	
Lectures	2 Hours/week	ESE	60 Marks
Practical	2 Hours/week	ISE	40 Marks
Credits	3	ICA	25 Marks

**Introduction:**

This course introduces Fundamentals and basic knowledge of an operating system. It also covers the details Process Management, deadlock, Memory Management, Unix System commands, and Introduction to Shell programming and awk programming

**Course Prerequisite:**

Students should have knowledge of Computer Systems and basics of C programming language.

**Course Objectives:**

1. To provide fundamental knowledge about operating system and process.
2. To explore process scheduling algorithm and synchronization techniques.
3. To understand methods for preventing or avoiding deadlocks in a computer system.
4. To examine memory management techniques for CPU performance.

**Course Outcomes:**

1. Comprehend the features of operating system to formulate its role, responsibilities and process concept.
2. Use different process scheduling algorithm and synchronization techniques.
3. Apply different methods for preventing or avoiding deadlocks in a computer system.
4. Analyze memory management techniques for CPU performance.

<b>Unit – I</b>	<b>Introduction</b>	<b>6 Hours</b>
Operating system definition, Simple Batch System, Multi-programmed Batch System, Time Sharing System, Personal Computer System, Parallel System, Real Time System		
<b>Unit – II</b>	<b>Process</b>	<b>7 Hours</b>
Process Concept, Process Scheduling, Operations on processes, Cooperating Processes, Threads, Inter-Process communication		
<b>Unit – III</b>	<b>Process Scheduling</b>	<b>6 Hours</b>
Basic concept, Scheduling Criteria, Scheduling Algorithms.		
<b>Unit – IV</b>	<b>Inter-process synchronization</b>	<b>7 Hours</b>
Background, The critical section problem, Peterson's algorithm, Semaphores, Classical problems of synchronization, Monitors.		
<b>Unit – V</b>	<b>Deadlocks</b>	<b>7 Hours</b>

System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock, combined approach to deadlock.		
<b>Unit – VI</b>	<b>Memory Management</b>	<b>7 Hours</b>
Background, Logical Versus Physical Address space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with paging.		
<b>Unit – VII</b>	<b>Virtual Memory</b>	<b>5 Hours</b>
Background, Demand paging, Page replacement, Page replacement algorithms, Allocation of frames, thrashing (Only concept).		
<b>Internal Continuous Assessment (ICA):</b>		
<b>Assignment List</b>		
<ol style="list-style-type: none"> <li>1. Implémentation of FCFS scheduling algorithm</li> <li>2. Implémentation of SJF (Préemptive and Non préemptive)</li> <li>3. Implémentation of round robin (RR).</li> <li>4. Implémentation of priority scheduling algorithm (non-preemptive)</li> <li>5. Implémentation of Mutual Exclusion using semaphore (wait &amp; signal)</li> <li>6. Implémentation Bankers Algorithm for Deadlock Avoidance</li> <li>7. Implémentation of RAG or WFG method for deadlock détection for single instance of ressources.</li> <li>8. Simulation of page replacement strategies (FIFO, LRU , Optimal)</li> <li>9. Simulation of Memory Allocation Strategies (First Fit, Best Fit, Worst Fit)</li> </ol>		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>1. Operating System concepts, Silberschatz, Galvin, 7 th or 8th Edition (John Wiley).</li> </ol>		



**23CSU5E14T : Programme Elective Course –I : MACHINE LEARNING**

Teaching Scheme		Examination Scheme	
Lectures	3 Hours/week	ESE	60 Marks
Practical	2 Hours/week	ISE	40 Marks
Credits	4	ICA	25 Marks

**Introduction:**

This course introduces Machine Learning which deals with fundamentals of machine learning and its types. The course also introduces designing of machine learning models by selecting appropriate machine learning algorithms for given problems. It also includes validation techniques, evaluation metrics, and hyper parameter tuning to evaluate and optimize the performance of machine learning models.

**Course Prerequisite:**

Prior knowledge of statistics, linear algebra, probability, and data pre-processing

**Course Objectives:**

1. To introduce various types of machine learning algorithms.
2. To enable designing of a model selecting appropriate machine learning algorithms for a given problem.
3. To introduce methods to evaluate and tune machine learning models.

**Course Outcomes:**

- At the end of the course students will be able to
1. Understand various types of machine learning algorithms and their applications.
  2. Design machine learning models by selecting appropriate machine learning algorithms for given problems.
  3. Evaluate and optimize the performance of machine learning models using validation techniques, evaluation metrics, and hyper parameter tuning.

<b>Unit – I</b>	<b>Introduction to Machine Learning</b>	<b>6 Hours</b>
What is Machine Learning? How do machine learn, Well-posed learning problem Types of Machine Learning: supervised learning, unsupervised learning, reinforcement learning, comparison – supervised, unsupervised and reinforcement learning, problems not to be solved using Machine Learning, applications of machine learning.		
<b>Unit – II</b>	<b>Supervised Learning- Classification</b>	<b>6 Hours</b>
Classification: Introduction, examples of supervised Learning, classification model, Classification learning steps, Common Classification Algorithms-Naïve Bayes classifier- conditional probability, Bayes theorem, applications of Bayes classifier, k-Nearest Neighbors (k-NN), decision tree, support vector machine, ensemble models.		
<b>Unit – III</b>	<b>Supervised Learning- Regression</b>	<b>6 Hours</b>

Regression: Introduction, Examples of regression, Common regression algorithms-Simple linear regression, Multiple linear regression, Assumptions in regression analysis, Main problems in regression analysis, Improving accuracy of the linear regression model, polynomial regression model, logistic regression model.		
<b>Unit – IV</b>	<b>Unsupervised Learning</b>	<b>8 Hours</b>
Introduction, Unsupervised vs Supervised learning, Applications of Unsupervised learning, Clustering: Clustering as a Machine learning task, different types of clustering techniques, Partitioning methods, k-means algorithm, k-medoids, Hierarchical clustering, Association rule, Apriori algorithm.		
<b>Unit – V</b>	<b>Validating Machine Learning Models</b>	<b>6 Hours</b>
Introduction: Bias, Variance, Underfitting, Overfitting, cost function, types of cost function, cross validation, Types of cross validation Training, Testing, Validation		
<b>Unit – VI</b>	<b>Modeling and Evaluation</b>	<b>6 Hours</b>
Introduction, selecting a Model, training a Model, Model, Representation and Interpretability, Evaluating Performance of a Model, Improving Performance of a Model. Applications and future of Machine learning.		
<b>Internal Continuous Assessment (ICA):</b>		
Minimum 8 assignments requiring students to develop machine learning applications for real world problem/use-case/scenario based on any of the following topics:		
<ol style="list-style-type: none"> <li>1. Linear and Multi-linear Regression</li> <li>2. Decision Tree regressor</li> <li>3. K-Nearest Neighbour (KNN) classifier</li> <li>4. Logistic Regression classifier</li> <li>5. Support vector machine (SVM)</li> <li>6. Decision Tree</li> <li>7. Naive-Bayes classifier</li> <li>8. Ensemble Models</li> <li>9. K-means clustering (Unsupervised Learning)</li> <li>10. Improving Machine Learning models using cross-validation and Hyper parameters tuning</li> </ol>		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>1. Machine Learning - Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, Pearson Publication.</li> <li>2. Machine Learning with Python for Everyone by Mark E. Fenner , Pearson Publication.</li> </ol>		
<b>Reference Books</b>		
<ol style="list-style-type: none"> <li>1. Machine Learning by Tom M. Mitchell (Publisher: McGraw Hill Education; First edition + New Chapters from Second edition).</li> <li>2. Introduction to Machine Learning (Second Edition) by Ethem Alpaydm (published by The MIT Press Cambridge, Massachusetts London, England</li> <li>3. Machine Learning and Data Science by Dr. Mayank R. Kothawade Dr. V S Giridhar Akula Dr. Pratibha C. Kaladeep (Yalagi), SIP International publishers.</li> <li>4. Machine Learning for Dummies by John Paul Mueller, Luca Massaron (Published by For Dummies; First edition).</li> </ol>		



**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
**(An Autonomous Institute)**  
**Third Year B.Tech. (Computer Science and Engineering), Semester-V**

**23CSU5E24T : Professional Elective I : DATA ANALYSIS AND VISUALIZATION**

Teaching Scheme		Examination Scheme	
<b>Lectures</b>	3 Hours/week	<b>ESE</b>	60 Marks
<b>Practical</b>	2 Hours/week	<b>ISE</b>	40 Marks
<b>Credits</b>	4	<b>ICA</b>	25 Marks
<b>Introduction:</b>			
In the field of data science, data pre-processing and visualization play a crucial role in preparing raw data for analysis and interpretation. This course covers fundamental techniques for cleaning, transforming, and visualizing data using various tools. Emphasis is placed on improving data quality and presenting insights effectively through graphical representations.			
<b>Course Prerequisite:</b>			
Fundamentals of Python Programming			
<b>Course Objectives:</b>			
<ol style="list-style-type: none"> <li>1. Understand the fundamentals and classifications of various types of data.</li> <li>2. Learn the principles and techniques of exploratory data analysis (EDA) and data pre-processing.</li> <li>3. Gain hands-on experience with NumPy and Pandas libraries for efficient data handling and manipulation.</li> <li>4. Learn to visualize data effectively using Python libraries like Matplotlib and Seaborn to communicate insights clearly.</li> </ol>			
<b>Course Outcomes:</b>			
At the end of this course, the student will be able to-			
<ol style="list-style-type: none"> <li>1. Identify the different types of data.</li> <li>2. Apply exploratory data analysis and pre-processing techniques for data preparation.</li> <li>3. Use NumPy and Pandas for data manipulation.</li> <li>4. Design the various charts using python libraries.</li> </ol>			
<b>Unit – I</b>	<b>Introduction and Describing Data</b>	<b>6 Hours</b>	
Overview, Sources of Data ,Process for Making Sense of Data, Observations and Variable, Types of Variables, Central Tendency, Distribution of the Data, Confidence Intervals, Hypothesis Tests			
<b>Unit – II</b>	<b>Preparing Data Tables</b>	<b>8 Hours</b>	
Overview, Cleaning the Data, Removing Observations and Variables, Generating Consistent Scales across Variables, New Frequency Distribution, Converting Text to Numbers, Converting Continuous Data to Categories, Combining Variables, Generating Groups, Preparing Unstructured, Data Visualizing Relationships between Variables, Calculating Metrics about Relationships.			
<b>Unit – III</b>	<b>Introduction to NumPy</b>	<b>8 Hours</b>	
Understanding Data Types in Python, The Basics of NumPy Arrays, Computation on NumPy Arrays: Universal Functions, Aggregations: Min, Max, and Everything in Between, Computation on Arrays: Broadcasting, Comparisons, Masks, and Boolean Logic, Fancy Indexing, Sorting Arrays, Structured			



Data: NumPy's Structured Arrays		
<b>Unit – IV</b>	<b>Data Manipulation with Pandas</b>	<b>7 Hours</b>
Installing and Using Pandas, Introducing Pandas Objects, Data Indexing and Selection, Operating on Data in Pandas, Handling Missing Data, Hierarchical Indexing, Combining Datasets: Concat and Append, Combining Datasets: Merge and Join, Aggregation and Grouping, Pivot Tables, Vectorized String Operations, Working with Time Series.		
<b>Unit – V</b>	<b>Data Visualization</b>	<b>6 Hours</b>
Overview, Visualization Design Principles, Tables, Univariate Data Visualization, Multivariate Data Visualization, Visualizing Groups, <b>Power BI</b> : Introduction, Installation of Power BI Desktop, Creating a Basic Visualization, Advanced Visualization and Customization.		
<b>Unit – VI</b>	<b>Visualization with Matplotlib and Seaborn</b>	<b>9 Hours</b>
General Matplotlib Tips, Simple Line Plots, Visualizing Errors, Density and Contour Plots, Histograms, Customizing Plot Legends, Customizing Colorbars, Multiple Subplots, Text and Annotation, Customizing Matplotlib: Configurations and Style sheets, Three-Dimensional Plotting in Matplotlib, Geographic Data with Basemap, Introduction to Seaborn: Seaborn functionalities and usage, Spatial Visualizations and Analysis in Python with Folium, Case Study.		
<b>Internal Continuous Assessment (ICA):</b>		
ICA should consist of minimum 8 practical assignments based on following topics:		
<ol style="list-style-type: none"> <li>1. Central Tendency and standard deviation</li> <li>2. Exploratory data analysis</li> <li>3. Data pre-processing techniques</li> <li>4. Data transformation techniques</li> <li>5. Time-series data analysis</li> <li>6. Data manipulation with NumPy</li> <li>7. Data analysis with Pandas</li> <li>8. Univariate and multivariate data visualization</li> <li>9. Design a Dashboard using PowerBI</li> <li>10. Three-Dimensional Plotting and special visualization</li> </ol>		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>1. Glenn J. Myatt, Making sense of Data: A practical Guide to Exploratory Data Analysis and Data Mining, John Wiley Publishers, 2014. (Unit- I and II)</li> <li>2. Glenn J. Myatt, Making sense of Data :A practical Guide to Data Visualization, Advanced Data Mining Methods and Applications, John Wiley Publishers, 2009.(Unit-V)</li> <li>3. Python Data Science Handbook–Essential Tools for working with Data: Jake Vander Plas, O’rielly (Unit III, IV, VI)</li> </ol>		



**23CSU5CC5P - ADVANCED JAVA PROGRAMMING**

Teaching Scheme		Examination Scheme	
Lectures	1 Hour/week	ISE	25 Marks
Practical	2 Hours/week	ICA	25 Marks
Credits	2	POE	50 Marks

**Introduction:**

It is an advanced technology or advance version of Java specially designed to develop web-based applications. It includes the concepts like Servlet, JSP etc. For developing web applications in Java , advanced Java fundamentals, like JSP, Servlets, JDBC etc. can add on to the capabilities and features of the application and thus are essential for developers.

**Course Prerequisite:**

Students should have knowledge of Core Java.

**Course Objectives:**

1. Enhance knowledge to manipulate and store data using JDBC
2. Inculcate skills necessary to develop web application using Java Servlet and Java Server Pages technology
3. Develop skills required to create java application using spring and hibernate framework

**Course Outcomes:**

Student will be able to:

1. Implement the web based applications using effective database access
2. Apply the concepts of server side technologies for creating dynamic web applications
3. Design web application using Spring framework.
4. Create a java application using Hibernate framework.

<b>Unit – I</b>	<b>Introduction to JDBC</b>	<b>3 Hours</b>
JDBC Introduction, JDBC features, JDBC Drivers, Setting up a database and creating a schema, Writing JDBC code to connect to DB, CRUD Operations with JDBC.		
<b>Unit – II</b>	<b>Servlets and Session Management</b>	<b>3 Hours</b>
Servlets Introduction, Need and Working, HTTP Methods; GET, POST, PUT, DELETE, TRACE, OPTIONS. GET/POST request; differences between the two, Servlet Lifecycle, ServletContext and ServletConfig, Forwarding and Redirection of requests.		
<b>Unit – III</b>	<b>Java Server Pages</b>	<b>2 Hours</b>
Introduction, Difference between JSP and servlet, Life cycle of JSP, JSP elements- Scriptlets, Expressions, Declarations, JSP Directives, Working with JSP basic tags and Implicit objects		
<b>Unit – IV</b>	<b>Spring Framework</b>	<b>4 Hours</b>
Introduction to Spring, Spring Architecture explanation and all its components, Spring MVC, Spring DAO, setting up of Spring framework Download JARs, Configure XML files.		

Unit – V	Hibernate Framework	3 Hours
ORM, ORM principle, ORM Implementation, Introduction to Hibernate, Hibernate Architecture, Persistent classes, Hibernate CRUD, setting up connection to DB using Hibernate		
<b>Internal Continuous Assessment (ICA):</b> Topics for Assignments <ol style="list-style-type: none"> <li>1. Command line application with JDBC</li> <li>2. GUI application for database access</li> <li>3. Java Login application</li> <li>4. Servlet application</li> <li>5. Servlet application for inserting data in database</li> <li>6. Servlet application for user authentication</li> <li>7. CRUD operations in JSP</li> <li>8. Application in JSP</li> <li>9. Spring application</li> <li>10. Hibernate application</li> </ol>		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>1. Java The Complete Reference, Ninth Edition, by Herbert Schildt, McGraw Hill Education</li> <li>2. Head First Servlets and JSP, Kathy Sierra, Bryan Basham, Bert Bates, O'Reilly Media, Inc.</li> </ol>		
<b>Reference Books</b>		
<ol style="list-style-type: none"> <li>1. Spring Microservices in Action, Second Edition, John Carnell, Manning Publications.</li> <li>2. Core and Advanced Java, Black Book, Dreamtech Press.</li> <li>3. Java Programming for Core and advanced learners, Sagayaraj, Denis, Karthik and Gajalakshmi, Universities Press</li> <li>4. Java Web Services: Up and Running, 2nd Edition, by Martin Kalin, O'Reilly Media, Inc.</li> </ol>		
<b>e-Resources</b>		
<ol style="list-style-type: none"> <li>1. <a href="https://hibernate.org/orm/documentation/6.0/">https://hibernate.org/orm/documentation/6.0/</a></li> <li>2. <a href="https://docs.spring.io/spring-framework/docs/current/reference/html/">https://docs.spring.io/spring-framework/docs/current/reference/html/</a></li> </ol>		



**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
(An Autonomous Institute)  
Third Year B.Tech. (Computer Science and Engineering), Semester-V

**23CSU5IK6T : Indian Knowledge System –II**

Teaching Scheme		Examination Scheme	
Lectures	2 Hours/week	ESE	-
Practical/Tutorial	-	ISE	50 Marks
Credits	2	ICA	-
<b>Introduction</b>			
Vedic Mathematics is a super-fast way of calculation. There are just 16 Sutras or Word Formulae which solve all known mathematical problems in the branches of Arithmetic, Algebra, Geometry, and Calculus. They are easy to understand, easy to apply, and easy to remember.			
<b>Course Prerequisite</b>			
Student shall have knowledge of basic mathematics.			
<b>Course Objectives</b>			
<ol style="list-style-type: none"><li>1. Introduce students to the foundational principles and principles of Vedic Mathematics.</li><li>2. Developmental calculation skills and enhance number sense.</li><li>3. Promote problem-solving abilities through mental and arithmetic techniques.</li><li>4. Build confidence in handling mathematical calculations quickly and accurately.</li><li>5. Foster interest in mathematics by making it more approachable and engaging.</li></ol>			
<b>Course Outcomes</b>			
At the end of the course, the students will be able to: <ol style="list-style-type: none"><li>1. Describe history and evolution of Vedic mathematics.</li><li>2. Perform Vedic arithmetic calculations with speed and accuracy.</li><li>3. Apply various Multiplication and Division methods to solve problems.</li><li>4. Calculate powers and roots of a number.</li><li>5. Solve simultaneous and quadratic equations.</li></ol>			
<b>Unit – I</b>	<b>Introduction of Vedic Mathematics</b>	<b>5 Hours</b>	
History and Evolution of Vedic Mathematics, Introduction of Basic Vedic Mathematics Techniques in Multiplication (Special Case, Series of 9, Series of 1 etc), Tables etc., Comparison of Standard Methods with Vedic Methods, sixteen sutras.			
<b>Unit – II</b>	<b>Vedic Arithmetic Operations</b>	<b>5 Hours</b>	
Various techniques to carry out basic operations covering Addition, Subtraction, Multiplication, Division, Complements and Bases, Vinculum number, General multiplication (Vertically Cross-wise).			

<b>Unit – III</b>	<b>Multiplication and Division</b>	<b>5 Hours</b>
Multiplications by numbers near base, Verifying answers by use of digital roots, Divisibility tests, Division of numbers near base, Comparison of fractions.		
<b>Unit – IV</b>	<b>Power and roots</b>	<b>10 Hours</b>
Applications of Vinculum, Different methods of Squares (General method, Base method, Duplex method etc.), Square roots, Cubes, Cube roots, General division		
<b>Unit – V</b>	<b>Vedic Algebra</b>	<b>5 Hours</b>
Simultaneous Equations, Quadratic Equations		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>1. Bhatiya Dhaval, Vedic Mathematics Made Easy, Jaico Publishing House</li> <li>2. Thakur Rajesh Kumar, Vedic Mathematics for students taking Competitive Examinations. Unicorn Books 2015 or Later Edition</li> <li>3. Gupta Atul, Power of Vedic Mathematics with Trigonometry, Jaico Books</li> <li>4. V. G. Unkalkar, Magical World of Mathematics (Vedic Mathematics), Vandana Publishers, Bangalore</li> <li>5. Vedic Math Genius by Kenneth Williams, Inspiration Books, England</li> </ol>		
<b>Reference Books</b>		
<ol style="list-style-type: none"> <li>1. Vedic Mathematics: Sixteen Simple Mathematical Formulae From The Vedas by Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja published by Motilal Banarsidass Publishers Pvt. Ltd., Delhi</li> <li>2. Lilavati: A Treatise Of Mathematics Of Vedic Tradition by Krishnaji Shankara Patwardhan published by Motilal Banarsidass Publishers Pvt. Ltd., Delhi</li> <li>3. The Power Of Vedic Maths by Atul Gupta, Jaico Publishing house, Mumbai</li> <li>4. Vertically And Crosswise By A. P. Nicholas, K. R. Williams, J. Pickles, Inspiration Books, Scotland</li> </ol>		



**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
(An Autonomous Institute)  
Third Year B.Tech. (Computer Science and Engineering), Semester-VI

**23CSU6CC1T - ARTIFICIAL INTELLIGENCE**

Teaching Scheme		Examination Scheme	
Lectures	3 Hours/week	ESE	60 Marks
Practical	2 Hours/week	ISE	40 Marks
Credits	4	ICA	25 Marks

**Course Objectives:**

1. To explore the foundational concepts, history, and real-world applications of Artificial Intelligence while understanding the structure and functionality of intelligent agents in different environments.
2. To develop problem-solving abilities using various search strategies and enhance knowledge representation and reasoning skills through logical and ontological approaches.
3. To express and analyze uncertainty through probabilistic models and Bayesian networks, as well as to grasp different learning paradigms for knowledge acquisition in artificial intelligence.

**Course Outcomes:**

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

1. Demonstrate an understanding of AI principles, its historical evolution, and the role of intelligent agents in decision-making across various domains.
2. Apply search algorithms and knowledge representation techniques to represent, analyze, and solve AI-related problems effectively.
3. Apply probabilistic reasoning model, including Bayes' Rule and Bayesian networks, to manage uncertainty and implement supervised, unsupervised, and reinforcement learning methods for decision-making.

<b>Unit – I</b>	<b>Overview of Artificial Intelligence</b>	<b>6 Hours</b>
What is AI-Definition, Foundations of AI (Philosophy, Mathematics, Neuroscience, Psychology, Computer Science), History, and Scope of AI, The state of the Art-Applications of AI in Real-World Scenarios.		
<b>Unit – II</b>	<b>Intelligent Agents</b>	<b>7 Hours</b>
Structure and Function of Intelligent Agents, Types of Agents: Simple reflex agent, Model based reflex agent, Goal-Based agent, Utility-Based agent, Learning Agents, The nature of Environment and types: Fully vs. Partially Observable, Real-World Examples of Intelligent Agents		
<b>Unit – III</b>	<b>Problem-Solving through Search</b>	<b>9 Hours</b>
State-Space Representation, Uninformed or blind Search Techniques: BFS, DFS, Informed or Heuristic Search Strategies: Greedy Best-First Search, A*, AO* Algorithm, Adversarial Search: Minimax Algorithm and Alpha-Beta Pruning.		
<b>Unit – IV</b>	<b>Knowledge Representation and Reasoning</b>	<b>7 Hours</b>
Logic, Propositional logic, First order logic: Representation, Syntax and Semantics: Models for First order logic, symbols & Interpretation, terms, Atomic sentences, Complex sentences quantifiers. Ontological-Engineering: Categories and objects, events.		

<b>Unit – V</b>	<b>Representing and Reasoning with Uncertain Knowledge</b>	<b>8 Hours</b>
Quantifying uncertainty: Acting under uncertainty: Summarizing uncertainty, Uncertainty and rational decisions, Basic Probability Notation: What probabilities are about, The language of propositions in probability assertions, Probability axioms and their reasonableness, Bayes' Rule and its use: Applying Bayes' rule: The simple case, Using Bayes' rule: Combining evidence, The semantics of Bayesian networks.		
<b>Unit – VI</b>	<b>Learning and Knowledge Acquisition</b>	<b>8 Hours</b>
Forms of Learning: Supervised, Unsupervised, Semi-supervised, Statistical learning, Reinforcement Learning, Q-learning, Sample applications.		
<b>Internal Continuous Assessment (ICA):</b> ICA will consist of <b>at least 10 programming assignments</b> using open-source AI tools and frameworks based on the following topics:		
<ul style="list-style-type: none"> <li>• Intelligent agents</li> <li>• Problem solving through search</li> <li>• First order logic</li> <li>• Bayesian Networks</li> <li>• Bayes' Rule</li> <li>• Learning and Knowledge Acquisition using Supervised and unsupervised algorithms</li> <li>• Statistical Learning</li> <li>• Q-learning.</li> </ul>		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>1. "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig, Prentice Hall.</li> <li>2. "A First Course in Artificial Intelligence" by Deepak Khemani, McGraw Hill Education (India).</li> <li>3. "Introduction to Artificial Intelligence &amp; Expert Systems" by Dan W. Patterson, PHI.</li> </ol>		
<b>Reference Books</b>		
<ol style="list-style-type: none"> <li>1. "Artificial Intelligence" by Elaine Rich and Kevin Knight, Tata McGraw Hill.</li> <li>2. "Machine Learning" by Tom M. Mitchell, McGraw Hill.</li> <li>3. "Pattern Recognition and Machine Learning" by Christopher M. Bishop, Springer.</li> </ol>		



**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
(An Autonomous Institute)  
Third Year B.Tech. (Computer Science and Engineering), Semester-VI

**23CSU6CC2T – SYSTEM SOFTWARE**

Teaching Scheme		Examination Scheme	
Lectures	3 Hours/week	ESE	60 Marks
Practical	2 Hours/week	ISE	40 Marks
Credits	4	ICA	25 Marks

**Introduction:**

This course introduces Language Processing activities, which helps to understand the basics of design and development of various professional languages, along with understanding of all the system software involved in executing a particular code written in a particular language.

A compiler translates a program written in a high-level programming language that is suitable for human programmers into the low-level machine language that is required by computers. Since writing a compiler is a nontrivial task, it is a good idea to split the compilation into several phases with well-defined interfaces. Conceptually, these phases operate in sequence, each phase except first phase taking the output from the previous phase as its input. Each phase is handled by a separate module.

This course provides an in-depth view of translation and optimization process. All phases required for translating a high-level language to machine language is covered in this course including scanning, parsing, intermediate-code generation, machine-code generation and code optimization.

**Course Prerequisite:**

- Theory of Computation
- Microprocessors
- Programming Language knowledge

**Course Objectives:**

1. To introduce basic principles of system software and tools.
2. To understand basic concepts, structure and design of assemblers, macro processors, linkers and loaders.
3. Understand fundamentals of compiler design and identify the relationships among different phases of the compiler.

**Course Outcomes:**

At the end of this course, the student will be able to -

1. Describe the basic principles of system software and implement assembler and macros to provide program generation facilities.
2. Distinguish between different loaders and linkers and their contribution in developing efficient applications.
3. Design and develop lexical analyzer and parser.
4. Produce collections of routines for generating intermediate code and apply code generation and optimization techniques on it.

<b>Unit – I</b>	<b>Language Processors</b>	<b>4 Hours</b>
-----------------	----------------------------	----------------

Introduction, Language processing activities, fundamentals of language processing, Fundamentals of language specification, language Processor development tools.

<b>Unit – II</b>	<b>Assemblers and Macro Preprocessors</b>	<b>10 Hours</b>
<p><b>Assemblers:</b> Elements of assembly language programming, a simple assembly scheme, Pass structure of assemblers, design of a two pass assembler.</p> <p><b>Macros and Macro Preprocessors:</b> Macro definition and call, Macro Expansion, Nested macro calls, Design of Macro pre-processor.</p>		
<b>Unit – III</b>	<b>Linkers and Loaders</b>	<b>8 Hours</b>
<p><b>Linkers:</b> Relocation and linking concepts, design of a linker, Self-relocating programs, linking for overlays. <b>Loaders:</b> Function of loader, General loader scheme, Absolute loader, Relocating loader, Direct linking loader, Dynamic loading, Design of direct linking loader.</p>		
<b>Unit – IV</b>	<b>Compilers: Lexical Analysis and Syntax Analysis</b>	<b>8 Hours</b>
<p>Introduction to Compiler, Phases of a compiler. <b>Lexical Analysis:</b> Role of a Lexical analyzer, Input buffering, Specification and recognition of tokens, Finite automata implications, designing a lexical analyzer generator. <b>Syntax Analysis:</b> Role of Parser, Top-down parsing, Recursive descent and predictive parsers (LL), Bottom-Up parsing, LR parsers, SLR parsers.</p>		
<b>Unit – V</b>	<b>Syntax Directed Translation and Intermediate Code Generation</b>	<b>7 Hours</b>
<p><b>Syntax Directed Translation:</b> Syntax directed definitions, construction of syntax tree, Bottom-up evaluation of S-attributed definitions, L-attributed definitions.</p> <p><b>Intermediate Code Generation:</b> Intermediate languages, declarations, Assignment statements, Boolean expressions, case statements, Backpatching.</p>		
<b>Unit – VI</b>	<b>Code Generation and Code Optimization</b>	<b>8 Hours</b>
<p><b>Code Generation:</b> Issues in design of a code generator and target machine, Basic blocks and flow graphs, Next use information and simple code generator.</p> <p><b>Code Optimization:</b> Sources of optimization, Peephole optimization and basic blocks, loops in flow graphs, Data flow analysis and equations, code improving transformation and aliases.</p>		
<p><b>Internal Continuous Assessment (ICA):</b></p> <ol style="list-style-type: none"> <li>Design Lex specifications for the tokens – keywords, identifiers, numbers, operators, and whitespaces.</li> <li>Design and Implementation of two pass assembler.</li> <li>Implementation of Macros.</li> <li>Implementation of Nested macros.</li> <li>Study of linkers.</li> <li>Study of loaders.</li> <li>Implement the lexical analyzer for simple ‘C’ language.</li> <li>Implement the shift- reduce parser.</li> <li>Implement the LL(1) parser for the language.</li> <li>Generate the symbol table for language.</li> <li>Generation of 3- address code for language and apply the code optimization technique on it.</li> <li>Generation of target code for given 3-address code.</li> </ol>		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>System Programming and operating systems – D.M. Dhamdhare 2nd Edition (TMGH)</li> <li>System Programming, J. J. Donovan, Mc-Graw Hill.</li> <li>Compilers - Principles, Techniques and Tools - A.V. Aho, R. Shethi and J.D. Ullman ( Pearson Education.)</li> </ol>		

## Reference Books

1. System Software- An Introduction to Systems Programming- 3rd Edition Leland L. Beck (Pearson Education)
2. Compiler Construction - Dhamdere (Mc-Millan)
3. Compiler Construction – Principles & Practice – Ken Loudon (Cengage Learning)
4. Compiler Design in C – Allen I. Holub (PHI / Pearson Education)
5. Compiler Construction: An advance course- Manish Kumar Jha (Dhanpat Rai) (3rd Edition)



**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
(An Autonomous Institute)  
Third Year B.Tech. (Computer Science and Engineering), Semester-VI

**23CSU6CC3T – SOFTWARE ENGINEERING**

Teaching Scheme		Examination Scheme	
Lectures	3 Hours/week	ESE	60 Marks
Credits	3	ISE	40 Marks
<b>Course Objectives:</b>			
1. To illustrate and compare the use of life cycle models for software development. 2. To identify different methods for analyzing and designing software requirements. 3. To apply different testing methods to the software system. 4. To describe various quality standards used in the software system.			
<b>Course Outcomes:</b>			
The Student will be able to 1. Select and apply the appropriate lifecycle model for software development. 2. Prepare SRS and SDS accordingly for a given problem. 3. Select and apply appropriate software testing method.			
<b>Unit – I</b>	<b>Introduction to Software Engineering</b>	<b>11 Hours</b>	
Introduction, The Problem Domain, Software Engineering Challenges and Approach, Software Process, Characteristics of Software Process, Software Development Process Models: Waterfall model, Prototype model, Iterative development model, Incremental model, Spiral model, Rational unified process model, Time boxing model, Agile process model.			
<b>Unit – II</b>	<b>Software Requirement Analysis &amp; Specification</b>	<b>6 Hours</b>	
Need of SRS, Characteristics of Good SRS, Requirement Process, Requirements specification, Functional Specification with Use Cases, DFD.			
<b>Unit – III</b>	<b>Software Architecture and Design</b>	<b>8 Hours</b>	
Introduction to Software Design, Software Architecture: Role of Software Architecture, Architecture Views, Component & Connector View, Design Concepts: Design Principles, Conceptual Design and Technical Design, Coupling, Cohesion, Function-Oriented Design, Object Oriented Design, High Level Design (UML Diagrams).			
<b>Unit – IV</b>	<b>Testing</b>	<b>4 Hours</b>	
Testing Fundamentals, Testing Process, Black-Box Testing, White-Box Testing, Object-Oriented Software testing methods, Functional testing			
<b>Unit – V</b>	<b>Project Planning and Management</b>	<b>8 Hours</b>	
Project management process, The Inspection and Audit Process, Software Configuration Management process, Effort estimation, Project Schedule and Staffing, Quality planning: Quality Concepts, Qualitative quality management planning. CMM project management process, Risk Management Planning, Project Monitoring plan			

<b>Unit – VI</b>	<b>Agile Project Management</b>	<b>8 Hours</b>
Introduction to APM, Implementation, Iterative Project Management Life Cycle, Adaptive Project Management Life Cycle, Scrum as a Agile methodology in project management.		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>1. An Integrated Approach to Software Engineering- 3rd edition: Pankaj Jalote (Narosa Publishers)</li> <li>2. Effective Project Management Traditional, Agile, Extreme, Robert K. Wysocki WILEY INDIA, 6<sup>th</sup> edition.</li> </ol>		
<b>Reference Books</b>		
<ol style="list-style-type: none"> <li>1. Ian Sommerville, software engineering, Pearson education Asia, 6th edition</li> <li>2. Software Engineering Fundamentals –Ali Behforooz and Frederick j. Hudson (Oxford University Press).</li> <li>3. Project Management with Scrum By Ken Schwaber</li> </ol>		



**Professional Elective-II : REINFORCEMENT LEARNING**

Teaching Scheme		Examination Scheme	
Lectures	3 Hours/week	ESE	60 Marks
Practical	2 Hours/week	ISE	40 Marks
Credits	4	ICA	25 Marks

**Introduction:**

Reinforcement learning is an area of machine learning, where an agent or a system of agents learns to archive a goal by interacting with their environment. In recent years there has been success in reinforcement learning research in both theoretical and applied fields. This course primarily focuses on training students to frame reinforcement learning problems and to tackle algorithms from dynamic programming, Monte Carlo, and temporal-difference learning.

**Course Pre-requisite:**

A basics of Machine learning, mathematical concepts of probability, linear algebra and statistics.

**Course Objectives:**

1. To provide a foundational understanding of the key concepts and components of Reinforcement Learning (RL).
2. To explore the methodologies and techniques used for decision-making and optimization in RL.
3. To introduce the core algorithms and approaches used in RL, focusing on their implementation and applications.
4. To develop an understanding of advanced RL techniques and their practical applications in real-world scenarios.

**Course Outcomes:**

At the end of the course students will be able to

1. Understand the fundamental concepts and components of Reinforcement Learning (RL).
2. Identify the methodologies and techniques used for decision-making and optimization in RL.
3. Describe the core RL algorithms and approaches, and recognize their applications in solving real-world problems.
4. Analyze advanced RL techniques and understand their effectiveness in real-world applications.

Unit – I	Introduction	7 Hours
Reinforcement Learning, Examples, Elements of Reinforcement Learning, History of Reinforcement Learning, The Agent–Environment Interface, Goals and Rewards, Returns, Unified Notation for Episodic and Continuing Tasks, State and action value Functions.		
Unit – II	Evaluative Feedback and Finite Markov Decision Processes	10 Hours
A k-armed Bandit Problem, Action-value Methods, The 10-armed Test-bed, Incremental Implementation.		
Unit – III	Dynamic Programming	5 Hours

Markov Property, Markov chain, Markov reward process (MRP), Introduction to Bellman equation for MRP, Markov decision process, (MDP), Bellman expectation equations, optimality of value function and policies, Bellman optimality equations.		
<b>Unit – IV</b>	<b>Temporal-Difference Learning</b>	<b>5 Hours</b>
TD Prediction, Advantages of TD Prediction Methods, SARSA: On-policy TD Control, Q-learning: Off-policy TD Control.		
<b>Unit – V</b>	<b>Planning and Learning</b>	<b>6 Hours</b>
Models and Planning, Dyna: Integrating Planning, Acting, and Learning, When the Model is wrong, Prioritized sweeping, Expected vs. Sample updates.		
<b>Unit – VI</b>	<b>Applications and Case Studies</b>	<b>6 Hours</b>
TD-Gammon, Samuel's Checkers Player, Mastering the Game of Go and AlphaGo, Deep Reinforcement Learning fundamentals and applications. RL- based Recommendation systems, resource management, and robotics.		
<b>Internal Continuous Assessment (ICA) :</b>		
Minimum 8 practical assignments-based on implementation of the following topics in Python.		
<ol style="list-style-type: none"> <li>1. K-armed Bandit Problem</li> <li>2. Incremental Implementation of Action-value Methods</li> <li>3. Markov Chain Simulation</li> <li>4. Bellman Equation for MDP</li> <li>5. Value Iteration</li> <li>6. Policy Iteration</li> <li>7. Temporal-Difference Learning (TD(0))</li> <li>8. SARSA (On-policy TD Control)</li> <li>9. Q-learning (Off-policy TD Control)</li> <li>10. Dyna-Q</li> <li>11. Simple game environments (CartPole, Mountain Car) using OpenAI Gym</li> <li>12. Building a recommendation system using contextual bandits</li> </ol>		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>1. Reinforcement Learning: An Introduction (Second edition + Upcoming Edition) by Richard S. Sutton and Andrew G. Barto, MIT Press Publication (The book is available at <a href="http://incompleteideas.net/book/the-book-2nd.html">http://incompleteideas.net/book/the-book-2nd.html</a> Upcoming edition's January 1 2018 draft available at <a href="http://incompleteideas.net/book/bookdraft2018jan1.pdf">http://incompleteideas.net/book/bookdraft2018jan1.pdf</a>]</li> <li>2. Hands-On Reinforcement Learning with Python: Master reinforcement and deep reinforcement learning using OpenAI Gym and TensorFlow by Sudharsan Ravichandran, Kindle Edition (Pact publishing)</li> </ol>		
<b>Reference Books</b>		
<ol style="list-style-type: none"> <li>1. Reinforcement Learning: With Open AI, Tensor Flow and Keras using Python by Abhishek Nandy, Manisha Biswas, Apress Publication.</li> <li>2. Reinforcement Learning: State-of-the-Art, Marco Wiering and Martijn</li> <li>3. Practical Deep Learning with Python by Pradeep Gohil (Apress publication) 1<sup>st</sup> edition.</li> </ol>		



**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
**(An Autonomous Institute)**  
**Third Year B.Tech. (Computer Science and Engineering), Semester-VI**

**Professional Elective-II : DISTRIBUTED SYSTEMS**

Teaching Scheme		Examination Scheme	
<b>Lectures</b>	3 Hours/week	<b>ESE</b>	60 Marks
<b>Practical</b>	2 Hours/week	<b>ISE</b>	40 Marks
<b>Credits</b>	4	<b>ICA</b>	25 Marks

**Course Objectives:**

1. To provide an Introduction to distributed Computing Systems, Distributed Operating systems and its design issues.
2. To describe various Communication techniques, message passing and Remote Procedure call mechanisms used for Inter process communication.
3. To discuss the synchronization issues such as Clock synchronization and mutual exclusion in distributed systems.
4. To study the issues and approaches for designing distributed file system.

**Course Outcomes:**

At the end of the course, student will be able to

1. Describe the design principles and message passing systems in distributed systems
2. Design and simulate communication models, Clock Synchronization and election algorithms in distributed system
3. Implement mutual exclusion distributed algorithms and analyze the design and functioning of Google file systems.

<b>Unit – I</b>	<b>Fundamentals</b>	<b>4 Hours</b>
Fundamentals of OS, What is Distributed System? Evolution of Distributed Computing System, Distributed Computing System Models, Distributed Computing Gaining Popularity, Issues in Designing Distributed System, Introduction to Distributed Computing Environment, Protocols for Distributed Systems – FLIP and VMTP		
<b>Unit – II</b>	<b>Message Passing</b>	<b>6 Hours</b>
Introduction, Desirable features of Good Message-Passing System, Issues in IPC by Message Passing, Synchronization, Buffering, Message Passing Interface, Multidatagram Messages, Process Addressing, Failure Handling, Group communication, Case Study: RMI, CORBA Unit		
<b>Unit – III</b>	<b>Remote Procedure Calls</b>	<b>6 Hours</b>
Introduction, The RPC Model, Transparency of RPC, Implementing RPC mechanism, Stub Generation, RPC Messages, Marshalling Arguments and Results, Server Management, Parameter-Passing Semantics, Call Semantics, Communication Protocols for RPCs, Client-Server Binding, Exception Handling, Security		
<b>Unit – IV</b>	<b>Synchronization in Distributed Systems</b>	<b>6 Hours</b>
Introduction, Process Migration, Threads, Clock Synchronization, Event Ordering, Election algorithms, Distributed Consensus algorithms		

<b>Unit – V</b>	<b>Distributed Mutual Exclusion</b>	<b>5 Hours</b>
Introduction, Classification of Mutual Exclusion Algorithms, Preliminaries, A simple solution To Distributed Mutual Exclusion, Non-Token-Based Algorithms, Lamport’s Algorithm, The Ricart-Agrawala Algorithm, Token-Based Algorithms, Suzuki-Kasami’s Broadcast Algorithms		
<b>Unit – VI</b>	<b>Distributed File Systems</b>	<b>6 Hours</b>
Introduction, Architecture, Mechanisms for building Distributed File System, Design issues, Log-Structured file systems, Case studies- Google FS		
<b>Internal Continuous Assessment (ICA) :</b> Minimum 8 to 10 assignments on the following topics		
<ol style="list-style-type: none"> <li>1. Client-Server communication model.</li> <li>2. Remote Procedure Call</li> <li>3. Election Algorithm</li> <li>4. Clock synchronization</li> <li>5. Token and Non-token based algorithms for mutual exclusion.</li> <li>6. Case Study of Google File System.</li> </ol>		
<b>Text Books</b>		
<ol style="list-style-type: none"> <li>1. Distributed O.S. Concepts and Design, P.K.Sinha, PHI</li> <li>2. Advanced Concepts in Operating Systems, Mukesh Singhal &amp; N.G.Shivaratri, TMH</li> <li>3. Distributed Computing, Sunita Mahajan, Seema Shah, OXFORD University Press</li> </ol>		
<b>Reference Books</b>		
<ol style="list-style-type: none"> <li>1. Distributed System Principles and Paradigms, Andrew S. Tanenbaum, 2nd edition, PHI</li> <li>Distributed Systems, Colouris, 3rd Edition</li> </ol>		



**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
**(An Autonomous Institute)**  
**Third Year B.Tech. (Computer Science and Engineering), Semester-VI**

**Professional Elective-III : NATURAL LANGUAGE PROCESSING**

Teaching Scheme		Examination Scheme	
Lectures	2 Hours/week	ISE	50 Marks
Credits	2		

**Introduction:**

Natural Language Processing (NLP) is essentially a method for teaching computers how to understand human languages and interpret text. This course covers fundamentals of NLP including language morphology and language modelling, syntax analysis, semantic analysis, and the use of NLP for information retrieval. It also covers the most popular and effective current techniques, strategies, and toolkits for NLP to develop the various real-world NLP applications.

**Course Pre-requisite:**

Basic mathematics, algorithms and programming skills, Theory of computation and parsing.

**Course Objectives:**

1. To provide students with a comprehensive understanding of the fundamental concepts, stages, and challenges in Natural Language Processing (NLP), along with foundational text processing techniques.
2. To enable students to analyze and understand the morphological and syntactic structures of language, and apply relevant linguistic models and parsing techniques for language interpretation.
3. To equip students with the skills to apply semantic analysis and language modeling techniques, helping them represent and interpret natural language meaningfully in various NLP applications.
4. To develop students' ability to utilize state-of-the-art NLP tools, techniques, and resources, enabling them to build practical applications such as information extraction, retrieval, and named entity recognition.

**Course Outcomes:**

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

1. Understand the basic concepts, stages, and challenges of Natural Language Processing, along with foundational text processing techniques.
2. Analyze morphological and syntactic structures of language using appropriate linguistic models and parsing techniques.
3. Apply semantic analysis and language modeling techniques to represent and interpret natural language meaningfully.
4. Utilize NLP tools, techniques, and resources to build applications involving information extraction and retrieval.

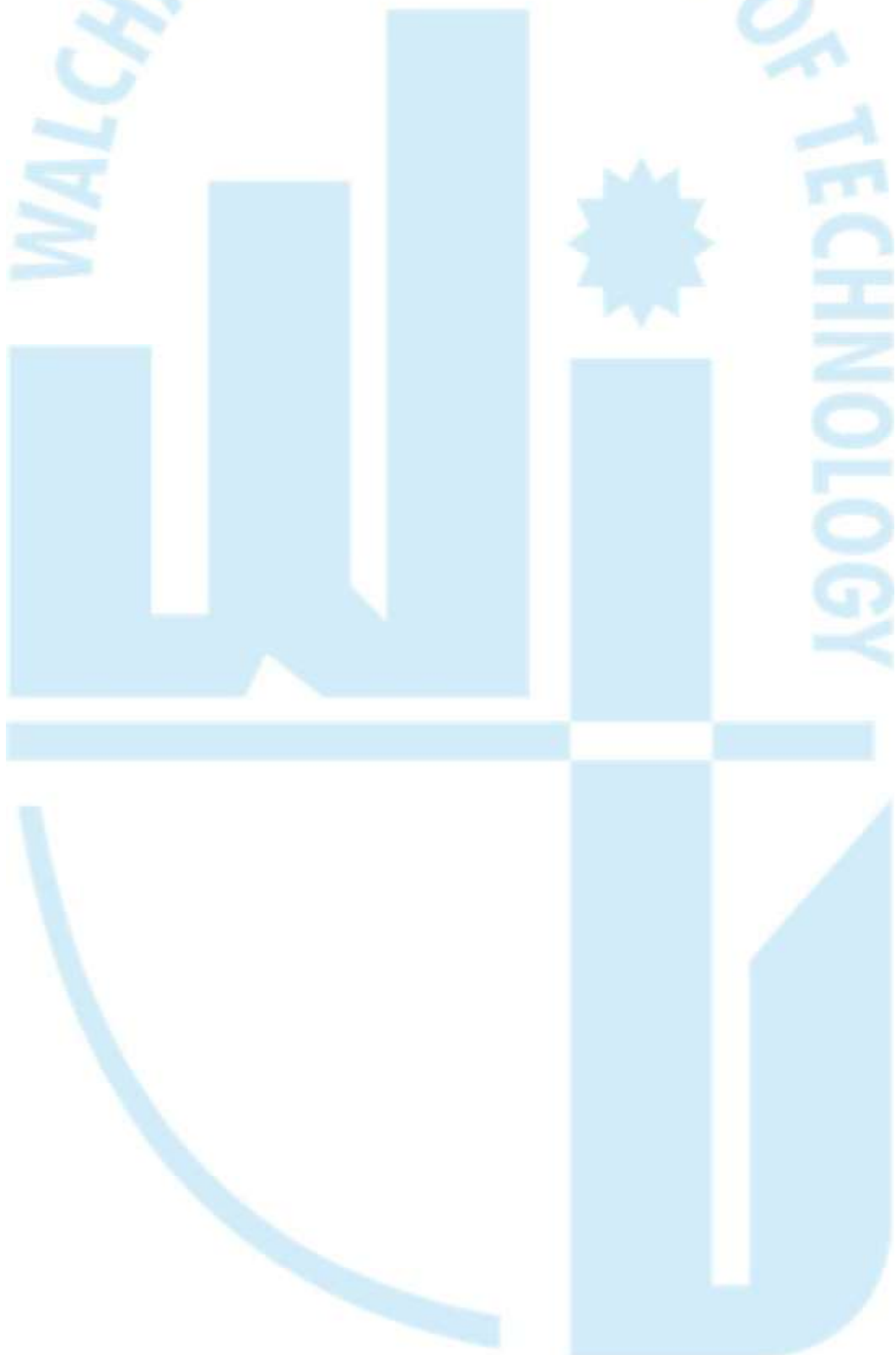
<b>Unit – I</b>	<b>Introduction to Natural Language Processing</b>	<b>4 Hours</b>
-----------------	--	----------------

Introduction to NLP, Machine Learning and NLP, Stages of NLP, challenges (Open Problems) in NLP.

**Basics of Text Processing:** Tokenization, Stemming, Lemmatization, Part of Speech Tagging.

<b>Unit – II</b>	<b>Morphology and Theory of Parsing</b>	<b>6 Hours</b>
Morphological Analysis: What is Morphology? Types of Morphemes, Inflectional morphology & Derivational morphology, Phonetics, HMM, Morphological Diversity of Indian Languages, Morphological parsing with Finite State Transducers(FSD). Syntactic Analysis: Syntactic Representations of Natural Language, Parsing Algorithms, Probabilistic context-free grammars and Statistical parsing.		
<b>Unit – III</b>	<b>Semantic Analysis</b>	<b>5 Hours</b>
Lexical Semantic, Relations among lexemes & their senses, Homonymy, Polysemy, Synonymy, Hyponymy, Word-Net, Word Sense Disambiguation (WSD), Lesk Algorithm Walker's algorithm, Word Nets for Word Sense Disambiguation, Dictionary based approach. <b>Case Studies:</b> Study of Stanford Parser and POS Tagger <a href="https://nlp.stanford.edu/software/lex-parser.html">https://nlp.stanford.edu/software/lex-parser.html</a>		
<b>Unit – IV</b>	<b>Language Modeling</b>	<b>6 Hours</b>
Probabilistic language modeling, Markov models, Generative models of language, Log-Liner Models, Graph-based Models. N-gram Models: Simple n-gram models, Estimation parameters and smoothing, evaluating language models. <b>Word Embeddings/Vector Semantics:</b> Bag-of-words, TFIDF, word2vec, doc2vec, Contextualized representations (BERT). <b>Topic Modeling:</b> Latent Dirichlet Allocation (LDA), Latent Semantic Analysis, Non –Negative Matrix Factorization.		
<b>Unit – V</b>	<b>Information Retrieval using NLP</b>	<b>4 Hours</b>
<b>Named Entity Recognition:</b> NER System building process, Evaluating NER System. Entity Extraction, Relation Extraction, Reference Resolution, Co reference resolution, Cross Lingua information retrieval (CLIR).		
<b>Unit – VI</b>	<b>NLP Tools, Techniques and Applications</b>	<b>5 Hours</b>
Prominent NLP libraries: Natural Language Toolkit (NLTK), spaCy, TextBlob, Gensim. <b>Linguistic Resources:</b> Lexical Knowledge Networks, WordNets, Indian Language WordNet, (IndoWordnet), Applications of NLP.		
<b>Text Books</b>		
1. Dr. Smita M. Chaudhari, Devika A. Verma, Nitin N. Sakhare, “Natural Language Processing”, Nirali Prakashan, Pune. 2. Allen, James, “Natural Language Understanding”, Second Edition, Benjamin/Cumming, 1995.		
<b>Reference Books</b>		
1. Bird, S., Klein, E., Loper, E. (2009). “Natural Language Processing with Python”. Sebastopol, CA: O'Reilly Media. 2. Akshay Kulkarni, AdarshaShivananda, “Natural Language Processing Recipes Unlocking Text Data with Machine Learning and Deep Learning using Python”, Apress, Bengalore. 3. Radford, Andrew et. al., “Linguistics, An Introduction”, Cambridge University Press, 1999. 4. Charniack, Eugene, “Statistical Language Learning”, MIT Press, 1993. 5. Joseph D. Booth Foreword by Daniel Jebaraj “Natural Language Processing Succinctly”, Succinctly Ebook series, SyncFusion.		

6. Github link for practical: <https://github.com/kb1907/Natural-Language-Processing-Specialization>





**WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR**  
(An Autonomous Institute)  
Third Year B.Tech. (Computer Science and Engineering), Semester-VI

**Professional Elective-III : PREDICTIVE ANALYTICS**

Teaching Scheme		Examination Scheme	
Lectures	2 Hours/week	ISE	50 Marks
Credits	2		

**Introduction:**

Predictive analytics is a branch of advanced analytics that uses historical data, statistical algorithms, and machine learning techniques to identify the likelihood of future outcomes. It helps organizations anticipate trends, behaviors, and events by analyzing current and past data patterns. By leveraging predictive analytics, businesses can make proactive, data-driven decisions to enhance efficiency and gain a competitive edge. It transforms raw data into valuable insights, enabling better planning and strategic actions across industries like finance, healthcare, marketing, and supply chain management.

**Course Pre-requisite:**

Predictive analytics requires technical and analytical skills. A strong foundation in statistics and probability is essential for understanding data patterns. Proficiency in programming languages like Python, R, or SQL is needed for data manipulation and model building. Knowledge of machine learning algorithms, such as regression and classification, is crucial. Skills in data cleaning, preprocessing, and visualization help in extracting meaningful insights.

**Course Objectives:**

1. To provide foundational knowledge of predictive analytics, including core concepts, techniques, and challenges related to handling diverse and complex datasets.
2. To expose students to real-world case studies from healthcare, social media, and marketing sectors to develop the ability to analyze, model, and interpret grouped data patterns.
3. To develop skills in data classification and preparation techniques essential for building accurate and reliable predictive models, including data cleaning, transformation, and dimensionality reduction.

**Course Outcomes:**

- At the end of this course students will be able to
1. Understand the core principles and techniques of predictive analytics, and demonstrate the ability to handle diverse and complex datasets effectively.
  2. Analyze real-world scenarios and case studies from healthcare, marketing, and social media domains to identify data patterns and apply suitable predictive models.
  3. Apply data classification methods and data preparation techniques such as cleaning, transformation, and dimensionality reduction to develop predictive solutions.

<b>Unit – I</b>	<b>Entering the Arena and Predictive analytics in the wild</b>	<b>5 Hours</b>
-----------------	--	----------------

**Entering the Arena:** Exploring predictive analysis, adding business values, starting predictive analytic project, ongoing predictive analysis.

**Predictive analytics in the wild:** Online marketing and retail, implementing a recommender system, Target marketing.

<b>Unit – II</b>	<b>Data types, associated techniques and complexities of data</b>	<b>5 Hours</b>
<p><b>Data types, associated techniques:</b> Recognizing your data types, Identifying data categories, generating predictive analytics, connecting to related disciplines.</p> <p><b>Complexities of data:</b> Finding value in your data, constantly changing your data, complexities in searching your data, exploration of raw data.</p>		
<b>Unit – III</b>	<b>Applying Models</b>	<b>5 Hours</b>
<p>Modeling data: Models and simulations, categorizing model. Healthcare analytics case studies: Google flu trends, Cancer survivability predictors, Social and marketing analytics case studies : Target store predicts pregnant women, Twitter-based predictors of earthquakes, Twitter-based predictors of political campaign outcomes, Twitter-based predictors of the stock market.</p>		
<b>Unit – IV</b>	<b>Identifying similarities in data</b>	<b>5 Hours</b>
<p>Explaining data clustering, Converting raw data into matrix: Creating a matrix of terms in documents, term selection, Identifying groups in data: K-means clustering algorithm, clustering by nearest neighbors, Finding Association in data items.</p>		
<b>Unit – V</b>	<b>Predicting the future using data classification</b>	<b>6 Hours</b>
<p>Explaining data classification, introducing data classification to business, exploring the data classification process, using data classification predict the future: Decision tree, algorithm for generating decision trees, support vector machines. Ensemble methods to boost prediction accuracy: naïve bayes classification algorithms, the markov model, linear regression, neural networks</p>		
<b>Unit – VI</b>	<b>Preparing data</b>	<b>4 Hours</b>
<p>Listing the objectives, Processing the data: Identifying the data, cleaning the data, generating the derive data, reducing the dimensionality of data, applying principal component analysis, structuring the data: extracting, transforming and loading the data, keeping the data up to date, outlining testing and test</p>		
<b>Text Books</b>		
<p>1. Predictive Analytics for dummies A Wiley brand: Anasse Bari, Mohamed Chaouchi, Tommy Jung. 2<sup>nd</sup> edition, published by John Wiley &amp; Sons.</p>		
<b>Reference Books</b>		
<p>1. Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die 2nd Edition, Kindle Edition by Eric Siegel (Author) Format: Kindle Edition</p> <p>2. Olivia Parr Rud “Data Mining Cookbook: Modeling Data for Marketing, Risk, and Customer Relationship Management”, Wiley,2001.</p> <p>3. Risk Management in Global Supply Chains", Wiley,2012.</p>		



**23CSVSEC6P : FULL STACK DEVELOPMENT**

Teaching Scheme		Examination Scheme	
Lectures	2 Hours/week	ISE	25 Marks
Practical	2 Hours/week	ICA	25 Marks
Credits	3	POE	50 Marks

**Introduction:**

Full stack web development course is designed to provide students with a solid foundation in the front-end and back-end web development fundamentals. The course is aimed at providing students with skills and knowledge in web development, including front-end and back-end development along with database access.

**Course Objectives:**

1. Inculcate skills necessary to design, develop and style a web-based user interface.
2. Develop ability to identify appropriate client/server-side scripting web technologies suitable for a given use case.
3. Develop skills required to create light weight, efficient and scalable browser-based APIs web applications.

**Course Outcomes:**

**At the end of this course students will be able to**

1. Structure and implement web pages using HTML, CSS and JavaScript.
2. Build interactive web pages using React JS and Express JS.
3. Develop web application using client/server-side scripting technologies.
4. Manage data using relational and non-relational database.

<b>Unit – I</b>	<b>Basics of HTML, CSS, and JavaScript</b>	<b>5 Hours</b>
HTML5: Features of HTML5, designing frontend using HTML5 CSS3: Features of CSS3, Styling frontend using CSS3 JavaScript: Syntax and Semantics of JavaScript, Document Object Model, Event Handling, Form handling and validations. JSON: Introduction to AJAX.		
<b>Unit – II</b>	<b>REACT JS</b>	<b>5 Hours</b>
Introduction, Fundamentals of React JS, working with Lists and Conditionals, Styling React Components and Elements, Adding Bootstrap in React, Debugging React apps		
<b>Unit – III</b>	<b>Express JS</b>	<b>5 Hours</b>
Introduction to Express JS, Views and Layouts, Middleware, Routing, Form Handling with Express, Request and Response Objects.		
<b>Unit – IV</b>	<b>Introduction to Server-side JS Framework – Node.js</b>	<b>5 Hours</b>
Introduction - What is Node JS, Architecture, Feature of Node JS, Installation and setup, creating web servers with HTTP (Request & Response), Event Handling, GET & POST implementation		

<b>Unit – V</b>	<b>Introduction to Mongo DB</b>	<b>5 Hours</b>
Getting Started with MongoDB Atlas, Overview of MongoDB and the Document Model, connecting to a MongoDB Database, MongoDB CRUD Operations: Insert, Find Documents, Replace, Delete, Modifying Query Results.		
<b>Unit – VI</b>	<b>PHP and MySQL</b>	<b>5 Hours</b>
Introduction to PHP 5 and PHP 6, variables and constants, program flow, functions, arrays and files and directories, Forms and Databases, integration with MySQL applications on PHP		
<b>In Semester Evaluation:</b> ISE Evaluation for the course will consists of three hands on tests based for 25 marks on the topics mentioned in the syllabus and a capstone project submitted in group of 4 students per group that will be evaluated for 10 marks.		
<b>Internal Continuous Assessment (ICA) :</b> Assignment Topics 1,2: HTML, CSS 3: Javascript 4: AJAX 5,6: React JS 7,8: Node JS 9: Express JS 10: Mongo DB 11: Mini project		
<b>Text Books / Reference Books</b>		
<ol style="list-style-type: none"> <li>1. Ben Frain, "Responsive Web Design with HTML5 and CSS3", Packt Publication</li> <li>2. Jon Duckett, "JavaScript and jQuery: Interactive Front-End Web Development"</li> <li>3. Official documentation of OpenAPI standard <a href="http://spec.openapis.org/oas/v3.0.3">http://spec.openapis.org/oas/v3.0.3</a></li> <li>4. Official documentation of Node.js: <a href="https://nodejs.org/en/">https://nodejs.org/en/</a></li> <li>5. Official documentation of React JS <a href="https://reactjs.org/docs/getting-started.html">https://reactjs.org/docs/getting-started.html</a></li> <li>6. Official documentation of PHP <a href="https://www.php.net/docs.php">https://www.php.net/docs.php</a></li> <li>7. Web link for MongoDB: <a href="https://www.mongodb.com/">https://www.mongodb.com/</a></li> <li>8. Official documentation of Express JS: <a href="https://devdocs.io/express-getting-started">https://devdocs.io/express-getting-started</a></li> </ol>		