Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology

Vidyanagari, Baramati, Dist. – Pune 413133
An Autonomous Institute Approved by AICTE and affiliated to SPPU,Pune

Department of Computer Engineering

Curriculum Structure and Syllabus of Final Year B. Tech Computer Engineering (Course 2023)

With effective from Academic Year 2026-27

INSTITUTE VISION AND MISSION

VISION

To achieve Academic Excellence through Persistent and Synergic Collaborations amongst all Stakeholders.

MISSION

1. To ensure holistic development of students as lifelong learners and problem solvers through value based quality education.

2. To motivate faculty to attain the state-of-the-art knowledge and wisdom in their domain and be a facilitator towards co- creation of knowledge

3. To frame and deploy conducive and empowering policies for multifaceted growth of students, faculty and staff to make them contributors towards excellence.

4. To partner with industry for mutually beneficial relations to generate employable and deployable workforce.

5. To fulfill the aspirations of alumni, parents, society, region and nation at large by generating technically competent.

DEPARTMENT VISION AND MISSION

VISION

To achieve excellence in the field of Computer Engineering with consistent and collaborative efforts of every individual

MISSION

1. To develop students with fundamental advanced tools and technologies to work as skilled Computer professionals with ethical values.

2. To promote faculty for higher education and expose them to current trends to enrich educational quality.

3. To provide appropriate environment with required resources to achieve academic excellence.

4. To develop hand-in-hand relations with industries for catering institute-industry needs.

5. To apply collaborative efforts to make students competent to provide solutions to social problems.

Program Specific Outcomes (PSO)

PSO1: Professional Skills

- Apply knowledge of algorithms, web design, databases, operating systems, data analytics, and networking to understand, analyze, and develop computer programs for the efficient design of computer-based systems with varying levels of complexity

PSO2: Problem-Solving Skills

- The capability to implement standard practices and strategies in software project development within open-ended programming environments, capability to utilize modern programming languages, environments, and platforms to build innovative career paths, pursue entrepreneurship, and foster enthusiasm for higher studies.

Program Educational Objectives (PEO)

1. Students will be able to apply the fundamentals, domain knowledge and modern technology of computer engineering to a provide effective and innovative solutions to engineering problems

2. Students will be able to solve societal challenging and multidisciplinary problems applying suitable resources

3. Students will be able to work as competent professional as an individual and a team member

Program Outcomes (POs)

PO1	**Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO2	**Problem Analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO3	**Design/Development of Solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO4	**Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO5	**Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO6	**The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO7	**Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
PO8	**Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO9	**Communication:** Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences
PO10	**Project Management and Finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO11	**Life-Long Learning:** Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology
Board of Studies: Computer Engineering
Syllabus: Final Year (B. Tech.) Computer Engineering
2023 Pattern w.e.f. AY:2026-2027

SEMESTER-VII

Course Type	Course Code	Course Name	Teaching Scheme			Examination Scheme and Marks							Credits			
			TH	PR	TUT	CAA	ISE	ESE	TW	PR	OR	Total	TH	PR	TUT	Total
IJOT	CO23401PR	Internship/On-job Training	-	24	-	100	-	-	100	-	150	350	-	12	-	12
EEMC	HS23401TH	Software Project Management	3	-	-	30	-	70	-	-	-	100	3	-	-	4
EEMC	HS23401TUT	Software Project Management	-	-	1	-	-	-	-	-	30	30	-	-	1	
RM	HS23403TH	Research Methodology	3	-	-	30	-	70	-	-	-	100	3	-	-	4
RM	HS23403TUT	Research Methodology	-	-	1	-	-	-	30	-	-	30	-	-	1	
AEC	HS23404TH	Public Specking and Aptitude	1	-	-	40	-	-	-	-	-	40	1	-	-	2
AEC	HS23404TUT	Public Specking and Aptitude	-	-	1	-	-	-	-	-	30	30	-	-	1	
Total			**7**	**24**	**3**	**200**	**-**	**140**	**130**	**-**	**210**	**680**	**7**	**12**	**3**	**22**

Dr. M. D. Shelar
Academic Coordinator

Dr. A. M. Jagtap
Head of Department

28/03/2026
Dr. S. M. Bhosle
Dean Academics

Dr. A. H. Kolekar
Controller of Examination

Dr. S. B. Lande
Principal

Principal
Vidya Pratishthan's
Kamalnayan Bajaj Institute of
Engineering & Technology, Baramati
Vidyanagari, Baramati-413133

VPKBIET BARAMATI
Affiliated to S. P. P. U. Pune Id No./PU/PW/Engg./152/2006
PUNE (MS.)

Vidya Pratishthan's
Kamalnayan Bajaj Institute of Engineering and Technology
Board of Studies: Computer Engineering
Syllabus: Final Year (B. Tech.) Computer Engineering
2023 Pattern w.e.f. AY:2026-2027
SEMESTER-VIII

Course Type	Course Code	Course Name	Teaching Scheme			Examination Scheme and Marks							Credits			
			TH	PR	TUT	CAA	ISE	ESE	TW	PR	OR	Total	TH	PR	TUT	Total
PCC	CO23411TH	Deep Learning	3	-	-	10	30	60	-	-	-	100	3	-	-	4
PCC	CO23411PR	Deep Learning	-	2	-	-	-	-	30	-	-	30	-	1	-	
PEC	CO23412TH	Program Elective Course-IV	3	-	-	10	30	60	-	-	-	100	3	-	-	4
PEC	CO23412PR	Program Elective Course-IV	-	2	-	-	-	-	-	-	30	30	-	1	-	
PEC	CO23413TH	Program Elective Course-V	2	-	-	10	-	60	-	-	-	70	2	-	-	3
PEC	CO23413PR	Program Elective Course-V	-	2	-	-	-	-	-	-	30	30		1	-	
MDM	X23XXXTH	Multi-disciplinary minor	2	-	-	20	20	50	-	-	-	90	2	-	-	3
MDM	X23XXXPR	Multi-disciplinary minor	-	2	-	-	-	-	20	-	-	20	-	1	-	
PROJ	CO23414PR	Project	-	8	-	-	-	-	80	-	50	130	-	4	-	4
OE	OE2300XX	Open Elective	2	-	-	-	-	50	-	-	-	50	2	-	-	2
	Total		12	16	0	50	80	280	130	-	110	650	12	8	-	20

List of Electives

Code	Program Elective-IV	Code	Program Elective-V
CO23412A	Natural Language Processing	CO23413A	DevOps Fundamentals
CO23412B	Soft Computing	CO23413B	Block Chain Technology
CO23412C	Software Architecture and Design Pattern	CO23413C	Quantum Computing

Dr. M. D. Shelar
Academic Coordinator

Dr. A. M. Jagtap
Head of Department

28/03/2026
Dr. S. M. Bhosle
Dean Academics

Dr. A. H. Kolekar
Controller of Examination

Dr. S. B. Lande
Principal

Principal
Vidya Pratishthan's
Kamalnayan Bajaj Institute of
Engineering & Technology, Baramati
Vidyanagari, Baramati-413133

VPKBIET BARAMATI
Affiliated to S.P.P.U. Pune
Id No. / PU / PN / Engg. / 152 / 2000
PUNE (MS.)

Multi-Disciplinary Minor and Open Elective

Multidisciplinary Minor (MDM) Subjects			
AI23051	AI & Machine Learning	ET23053	Internet of Things
AI23052	Data Science	CE23051	Waste Management
AI23053	Generative AI	CE23052	Green Building & Smart Cities
CO23051	Cloud Computing	ME23051	Introduction to 3D Printing Technologies
CO23052	High Performance Computing	ME23052	Introduction to Robotics & Automation
CO23053	Computer Graphics & Gaming	EL23051	Solar Tech
IT23051	Cyber Security	EL23052	Industrial Automation
IT23052	Full Stack Development	GS23051	Nano Technology
ET23051	Embedded Systems	GS23052	Linear Algebra and Statistics
ET23052	Drone Technology		

Open Electives (OE) Subjects			
OE23001	Digital Marketing	OE23011	Biotechnology
OE23002	Professional Leadership	OE23012	International Relations
OE23003	Organizational Behavior	OE23013	Universal Human Values
OE23004	Industrial Management	OE23014	Education Technology
OE23005	Disaster Management	OE23015	Design Thinking
OE23006	Energy Economic & Management	OE23016	Financial Literacy for Bharat#
OE23007	Operation Research	OE23017	Sustainability & Climate Change
OE23008	Intellectual Property Rights	OE23018	Agriculture Technology
OE23009	Cyber Laws	OE23019	Architectural Technology
OE23010	Bioinformatics		

Semester VII

CO23401- INTERNSHIP		
Teaching Scheme:	**Credits: 12**	**Examination Scheme:**
TH: 00 Hrs./Week **PR: 24 Hrs./Week**	**Theory: 00** **Practical : 12**	**CAA:**100 **Term Work :**100 **Lab Evaluation:** 150 Marks

Prerequisites:

Students should have successfully completed **Semester VI** of the Engineering program.

Course Objectives:

- To learn and practice hands-on technical skills.
- To provide opportunities for acquiring, comprehending, and refining practical technical proficiencies.
- To gain exposure to professional industrial practices and environments.
- To understand how real-world factors like cost, society, and management affect a company.
- To cultivate ethical principles aligned with professional and societal standards.

Course Outcomes:

CO1: Understand how people in the industry approach and solve problems.

CO2: Learn how to write clear and effective technical reports.

CO3: Work effectively and professionally as part of a team.

CO4: Learn to pick the right tools and technology to solve a given problem.

CO5: Analyze various career opportunities and decide career goals.

CO6: Demonstrate abilities of a responsible professional and use ethical practices in day-to-day life.

Internship work Identification:

Student may choose to undergo Internship at Industry/Govt. Organizations/NGO/MSME/Rural Internship/ Innovation/IPR/Entrepreneurship. Student may choose either to work on innovation or Entrepreneurial activities resulting in start-up or undergo internship with industry/NGO's/Government organizations/Micro/Small/ Medium enterprises to make themselves ready for the industry.

Internship work identification process should be initiated in the VI semester in coordination with training and placement cell/ industry institute cell. This will help students to start their internship work on time.

Student can take internship work in the form of the following but not limited to:

1. Working for consultancy/ research project
2. Contribution in Incubation/ Innovation/ Entrepreneurship Cell/ Institutional Innovation Council/ startups cells of institute /Learning at Departmental Lab/ Institutional workshop
3. Development of new product/ Business Plan/ registration of start-up
4. Industry / Government Organization Internship
5. Internship through Internshala,
6. In-house product development, intercollegiate, inter department research internship under research lab/group, micro/small/medium enterprise/online internship,
7. Research internship under professors, IISC, IIT's, Research organizations,
8. NGOs or Social Internships, rural internship,
9. Participate in open source development.

Duration:
Internship is to be completed after semester 6 and before commencement of semester 8 of at least 14 to 20 weeks; and it is to be assessed and evaluated in semester 7.

Guidelines for students

1. All B.Tech students are required to undergo an internship for a minimum duration of **14–16 weeks**. Students must **obtain a Final Year - Bonafide Certificate through the college office**, which is **mandatory** for commencing the internship.
2. Student must submit application form with all documents and Undertaking forms to department IIIC Coordinator through mentor
3. Students can take mini projects, assignments, case studies by discussing it with concerned authority from industry and can work on it during internship.
4. All students should compulsorily follow the rules and regulations as laid by industry.
5. Every student should take prior permissions from concerned industrial authority if they want to use any drawings, photographs or any other document from industry.
6. Student should follow all ethical practices and SOP of industry.
7. Students have to take necessary health and safety precautions as laid by the industry.
8. Student should contact his /her academic guide from college on weekly basis to communicate the progress.
9. Each student has to prepare internship report in consultation with the academic guide.

Internship Report

1. Students must prepare a **comprehensive internship report** in the **format prescribed by the department**, covering observations, tasks performed, and key learning outcomes.
2. Students may consult the **Industry Supervisor** and **Faculty Mentor** to obtain specific topics or problem statements to be included in the report.
3. Students should use the **daily diary** as a reference while preparing the report, as it already contains detailed information recorded during the internship.
4. The completed report must be **signed by the Industry Supervisor and Faculty Mentor** before submission to the department.
5. The internship report will be evaluated based on the following criteria:
    - Originality of content
    - Adequacy and purposefulness of the write-up
    - Organization, formatting, quality of sketches/drawings, writing style, and language
    - Variety and relevance of the learning experiences documented
    - Linkage of practical applications with theoretical concepts learned in coursework

Internship Diary / Internship Workbook

1. Students must maintain a daily diary **in the format prescribed by the college**, documenting observations, tasks performed, information gathered, and any suggestions.
2. The diary should include relevant sketches, drawings, or diagrams based on daily observations.
3. The industry supervisor or section in-charge must **verify and sign the diary every week**.
4. Students must present the diary to the **Faculty Mentor during each industry visit** for verification and ratification.
5. Upon completion of the internship, students must submit the following to the Institute:
    - Student's Daily Diary (as per college format)
    - Internship Report

- Attendance Record
- Evaluation Sheet duly signed and stamped by the industry

The diary will be evaluated based on **regularity of entries, completeness, and adherence to the prescribed format**.

Evaluation Guidelines:

Every student is required to prepare and maintain documentary proofs of the activities done by him/her as internship diary or as workbook. The evaluation of these activities will be done by Department IIIC Coordinator / faculty mentor or Industry Supervisor/Appointed External Examiner based on- Overall compilation of internship activities, sub-activities, the level of achievement expected, evidence needed to assign the points and the duration for certain activities. Assessment and Evaluation is to be done in consultation with internship supervisor (Internal and External – a supervisor from place of internship.)

Component	Marks
A. Continuous Assessment Activity	100
B. Term-Work (Internship Report)	100
C. Oral Examination / Viva	150
TOTAL	350

1. **Continuous Assessment (100 Marks):** Evaluation includes attendance, discipline, workplace behavior, and the quality of the learning diary or logbook. Mid-semester progress presentations and structured industry supervisor feedback form an integral part of the continuous review process.
2. **Term-Work (150 Marks):** Assessment covers the structure and completeness of the internship report, technical depth, problem-solving ability, and reflection on skills developed. Supporting evidence—drawings, screenshots, certificates, and attendance records—is verified, along with a plagiarism check to ensure originality.

3. **Oral Examination / Viva (150 Marks):** Evaluation focuses on understanding of tasks performed, application of engineering concepts, clarity of communication, industry relevance, and the ability to respond logically during interaction with examiners. Joint assessment by internal and external examiners ensures transparency and fairness.

Internship Evaluation Scheme (Total: 350 Marks)

Component	Marks	Evaluation Basis	Mode of Assessment / Evaluator
A. Continuous Assessment (50 Marks)			
Attendance, Discipline & Professionalism	20	Regularity, punctuality, adherence to workplace culture	Attendance record + Industry Supervisor note
Diary / Logbook	20	Weekly reflection of tasks, learning outcomes, challenges	Logbook review by Faculty Mentor
Mid-Semester Progress Presentation	20	Presentation on tasks performed, tools/technologies learned, contributions	Faculty review (in consultation with Industry Supervisor)
Industry Supervisor Feedback	40	Attitude, initiative, teamwork, professional conduct	Structured feedback form
Subtotal (A)	**100**		
B. Term-Work (Internship Report)- (100 Marks)			
Internship Report (Structure & Completeness)	20	Cover page, acknowledgement, organization profile, objectives, methodology, tasks, outcomes, conclusion	Faculty Panel Evaluation
Technical Content & Problem Solving	30	Depth of technical work, relevance to discipline, engineering application	Faculty Panel
Skill Development Reflection	20	Technical/professional skills, tools learned, employability skills (NEP focus)	Faculty Panel
Evidence & Annexures	15	Screenshots, codes, drawings, certificates, datasheets, attendance logs	Faculty Panel
Plagiarism / Originality	15	Minimum 80% originality; no copy-paste	Plagiarism check + Faculty review
Subtotal (B)	**100**		
C. Presentation & Oral Examination / Viva (100 Marks)			
Understanding of Work Done	30	Explanation of tasks performed, process understanding, technical depth	Internal + External Examiners
Application of Knowledge	30	Linkage with curriculum, application of engineering concepts	Viva Panel
Soft Skills & Communication	30	Presentation skills, clarity, professional conduct	Viva Panel
Industry Relevance & Employability	30	Awareness of industry practices, teamwork, adaptability	Viva Panel (Industry input)
Q&A Interaction	30	Logical reasoning and accuracy of responses	Viva Panel
Subtotal (C)	**150**		
Total A+B+C	**350**		

Feedback from internship supervisor

Post internship, faculty coordinator/Mentor should collect feedback about student with recommended parameters include as- Technical knowledge, Discipline, Punctuality, Commitment, Willingness to do the work, Communication skill, individual work, Team work, Leadership

Reference:

1. https://internship.aicte-india.org/
2. Circular No. 29-2024 Internship Cell- BOD Link_15022024.pdf
3. https://www.aicte-india.org/sites/default/files/AICTE%20Internship%20Policy.pdf

HS23403: Research Methodology and IPR

Teaching Scheme:	Credit: 04	Examination Scheme:
TH : 03 Hrs/Week TUT:01 Hrs/Week	TH Credit :03 TUT Credit :01	Course Activity: 30 Mark End Semester:70 Mark Term work : 30 Mark

Prerequisite: Project based learning of all subjects, Fundamental laws and principles of all subjects, Soft and communication skills.

Course Objective:

- To know people management in software project1. The course has been developed with orientation towards research related activities and recognizing the ensuing knowledge as property.
- It will create consciousness for Intellectual Property Rights and its constituents.
- Learners will be able to perform documentation and administrative procedures relating to IPR in India as well as abroad.

Course Outcomes:

On completion of the course, the students will be able to:

1. Formulate a research problem for engineering and technology domain.
2. Analyze the available literature for given research problem and understand different techniques of data collection.
3. Investigate the statistical and reliability methods of preliminary data analysis and present the results in graphical form.
4. Understand the importance of technical writing and presentation skills.
5. Comprehend the various forms of the intellectual property, its relevance and business impact in the changing global business environment.
6. Realize the importance of patents, trademark and copyright and follow research ethics.

Course Contents

Mapping of Course Outcomes for Unit I	CO1

UNIT I	Introduction	07 Hours

Introduction, Meaning of research, Objectives of research, Types of research, Research approaches, Significance of research, Research methods versus methodology, Research and scientific method, Research process, Criteria of good research, Problems encountered in India for good research, Formulation of research hypotheses, Search for causation, Format for research proposal, Funding for the proposal, Different funding agencies, Framework for the planning.

Mapping of Course Outcomes for Unit II	CO2

UNIT II	Literature Review	07 Hours

Definition of literature and literature survey, Significance of literature survey, Sources of literature, Elements and objectives of literature survey, Styles of literature survey, Strategies of literature survey, Searching the existing literature, Reviewing the selected literature, Writing about the literature reviewed and gap identified, literature analysis, data collection, and interpretation.

Mapping of Course Outcomes for Unit III		CO3
UNIT III	**Preliminary Data Analysis**	**07 Hours**
Testing of hypothesis- concepts and testing, Review of theory of reliability, Hazard models, System reliability. Data presentation skills, Features of statistical analysis, Histogram, bar charts, Pie charts, 2D & 3D plots, Interpolation & extrapolation techniques, Curve fitting.		
Mapping of Course Outcomes for Unit IV		**CO4**
UNIT IV	**Technical Writing and Presentation**	**07Hours**
Effective technical writing, thesis writing, research proposal writing, research paper writing. Significance of report writing, Different steps in writing report, Layout of the research report, Types of reports, Mechanics of writing a research report, Precautions for writing research reports, Presentation skills, tools for technical writing and presentation. Plagiarism, avoiding plagiarism, Research ethics, Tools for plagiarism checking, technical writing and presentation.		
Mapping of Course Outcomes for Unit V		**CO5**
UNIT V	**Intellectual Property Rights**	**07 Hours**
Introduction and significance of intellectual property rights, Types of Intellectual Property Rights, Copyright and its significance, Introduction to patents and its filing, Introduction to patent drafting, Best practices in national and international patent filing, Copyrightable work examples.		
Mapping of Course Outcomes for Unit VI		**CO6**
UNIT VI	**Patent Rights**	**07 Hours**
Patents and its basics, Patentable items, Designs, Process of filing patent at national and international level, Process of patenting and development, Technological research and patents, innovation, Patent and copyright international intellectual property, Procedure for grants of patents, Need of specifications, Types of patent applications, Provisional and complete specification, Patent specifications and its contents, Trade and copyright.		

Books and Other Resources

Text Books:

1. Ranjit Kumar (2005), 2nd edition, Research Methodology: A Step by Step Guide for beginners (Pearson Education).
2. C. K. Kothari (2004), 2nd edition, Research Methodology Methods & Techniques (New Age International, New Delhi).
3. T. Ramappa (2016), 2nd edition, Intellectual Property Rights-Law in India (Asia Law House, Hyderbad).

Reference Books:

1. Louis Cohen, Manion, Morrison and Routledge (2017), 8th edition, Research Methods in Education (Taylor & Francis Group- Cambridge University Press India Pvt. Ltd.).
2. John Best and James Kahn (1998), 8th edition, Research in Education (Prentice Hall of India Pvt. Ltd.).
3. Stuart Melville and Wayne Goddard (2001), Research Methodology: An Introduction for Science and Engineering Students. (Juta & Co Ltd.).
4. Benjamine Niebel and Alan Draper (1974), Product Design and Process Engineering, (McGraw Hill International Publishers).

4. Halbert D. J. (2007), 2nd edition, Resisting Intellectual Property (Taylor and Francis Ltd.).

5. Robert P. Merges, Peter S. Menell, Mark A. Lemley (2016), Intellectual Property in New Technological Age (Stanford Public Law Working Paper No. 2780190, Elsevier Publishers).

The term work should consists of following and the oral examination is conducted based on following assignments.

1. **Literature Review:** Collect the existing literatures on any research idea in engineering/technology and find out the research gap**. (Performed in a group of students of not more than three).**
2. **Report and Seminar Presentation:** Prepare the research proposal based on the earlier identified research gap (Report should check in plagiarism and grammarly) and present the idea. **(Performed in a group of students of not more than three).**
3. **Blank format of research proposal:** Identify the national and international funding agencies and prepare/print the blank format of research proposal of any one funding agency. **(Performed in a group of students of not more than three).**
4. Write a report on different citation style and reference style adopted by different publishers.
5. Write a report on case study of any existing patent/copy right/trademark.
6. Collect the information of any one referred peer reviewed journal and write a report based on Abstracting and indexing, H Index, SJR rating, Impact factor, Aims and Scope of the Journal, Guidelines for paper submission etc.

HS23401 : Software Project Management

Teaching Scheme:	Credit: 04	Examination Scheme:
TH: 03 Hrs/Week TUT: 01 Hrs/Week	TH Credit :03 Oral Credit :01	Course Activity:30 Mark End Semester: 70 Mark Oral : 30 Mark

Prerequisite: Students are expected to have a good understanding of Software Engineering

Course Objective:

- To understand the fundamentals of Software Project Management
- To investigate software project planning and management tools
- To learn software project scheduling and tracking
- To discuss about the agile project management
- To know people management in software project

Course Outcomes:

1. Comprehend Project Management Concepts
2. Use various tools of Software Project Management
3. Schedule various activities in software projects
4. Track a project and manage changes
5. Apply Agile Project Management
6. Analyse staffing process for team building and decision making in Software Projects and Management

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Mini Project
2. Survey on uses of software project management with emerging technology presentation
3. Industry Visit
4. Seminar
5. Research Paper in software project management

Course Contents

Mapping of Course Outcomes for Unit I	CO1

UNIT I	Introduction to Software Project Management	07 Hours

Project Definition, Project versus Flow type work, Project Lifecycle, Processes and Knowledge Areas in Project Management (PM), Build or Buy decision, Work Breakdown Structure (WBS) and its types, Introduction to PMBOK, Program and Portfolio Management. Case Studies: Analysis of a project using project management Concepts

Mapping of Course Outcomes for Unit II	CO2

UNIT II	Project Planning and Project Management Tools	07 Hours

Project Planning: Steps for Project Planning, PERT and Gantt Charts, Gantt Project, Microsoft Project and Primavera Project Management Software, Objectives of Activity planning, Project Schedules, Activities,

Sequencing and Scheduling, Network Planning Models, Formulating Network Model. Case Studies: Create software project plan using any tool.

Mapping of Course Outcomes for Unit III	CO3

UNIT III	Activity Based Scheduling	07 Hours

Introduction, Objectives of Activity Planning, Project Schedules: Activities: Sequencing and Scheduling, Network Planning Models, Formulating Network Model, Activity relationships (FS, SF, SS, FF), Forward Pass and Backward Pass techniques, Critical Path concept and remedies. Case Studies: Apply the critical path technique to the project

Mapping of Course Outcomes for Unit IV	CO4

UNIT IV	Software Project Monitoring and Control	06 Hours

Introduction, Collection of Project data, Visualizing progress, Cost monitoring, Earned Value Analysis, Project tracking, Change Control, Software Configuration Management, Managing contracts, Contract Management. Case Studies: Analyse the effect of a major requirement change on the schedule.

Mapping of Course Outcomes for Unit V	CO5

UNIT V	Agile Project Management	06 Hours

Predictive versus Empirical Management, Comparison between Non-Agile and Agile Project, Three stages of Agile Project, Estimation, Scope Management, Roles and Responsibilities, Scheduling and Tracking.
Case Studies : Analyse the same project using Agile. Create the three stages of the project.

Mapping of Course Outcomes for Unit VI	CO6

UNIT VI	Staffing in Software Projects	06 Hours

Managing People, Organizational behaviour, Best methods of Staff Selection, Motivation, The Oldham, Hackman job characteristic Model, Stress, Health and Safety, Ethical and Professional concerns, Working in Teams, Decision Making, Organizational structures, Dispersed and Virtual Teams, Communications Genres, Communication Plans. Case Studies: Analyse a case study for a distributed team and comment.

Books and Other Resources

Text Books:

1. Bob Hughes, Mike Cotterell and Rajib Mall, Software Project Management, Sixth Edition, Tata McGraw Hill, New Delhi, 2017
2. Robert K. Wysocki, Effective Software Project Management, Wiley Publication, 2011

Reference Books:

1. Ken Schwaber, Agile Project Management, Microsoft Press, 2004
2. Walker Royce, Software Project Management, Addison-Wesley, 1998
3. Jalote Pankaj, Software Project Management in Practice, Addison-Wesley Professional, 2002
4. PMBOK Guide

e Books:

1. https://www.konve-online.net/ITIL/McgrawHill.Software_Project_Management_2nd_Edition.pdf
2. http://library.lo/main/B96E3B1223262FBD62C6FDB35A5E978422

HS23404: Public Speaking And Aptitude

Teaching Scheme: TH : 01 Hrs/Week PR:02 Hrs/Week	**Credit: 02** TH Credit :01 PR Credit :01	**Examination Scheme:** Course Activity:40 Mark Oral : 30 Mark

Prerequisite: Pre – requisites: Communication and Professional Skills

Course Contents

Mapping of Course Outcomes for Unit I

UNIT I	**Spoken English**	**04 Hours**

Pre-Assessment, Vocabulary made easy, the Power of Words , Introduction to Word Accent: Disyllabic Words, Shift of Accent in Disyllabic Words, Polysyllabic Words, Introduction to Rhythm : Intonation, Rising Intonation, Falling Intonation, Introduction & Specific scenarios: Telephone Skills: Taking & Making Calls, Voice, Intonation, and Language, Conversations: The Role of Questions, Types and Functions of Questions, avoiding Counterproductive Questions

Mapping of Course Outcomes for Unit II

UNIT II	**Impactful Presentations:**	**04 Hours**

Body Language: Introduction, Mechanics & Style Voice Modulation: Voice Projection, replacing Fillers, and Emphasis Power of Pause: Pause to engage audience in Conversation, Combine Pause & Repetition Techniques, Demonstrate Confidence & Control, establish Presence Empathy: Essential Human Quality, Practice Heartful Communication, Impact of Communication, How to deliver memorable speech.

Mapping of Course Outcomes for Unit II

UNIT III	**General Aptitude for all Competitive ExamsVerbal Aptitude**	**06 Hours**

Verbal Aptitude

Basic English grammar: tenses, articles, adjectives, prepositions, conjunctions, verb-noun agreement, and other parts of speech Basic vocabulary: words, idioms, and phrases in context, reading and comprehension, Narrative sequencing.

Quantitative Aptitude

Data interpretation: data graphs (bar graphs, pie charts, and other graphs representing data), 2- and 3-dimensional plots, maps, and tables Numerical computation and estimation: ratios, percentages, powers, exponents and logarithms, permutations and combinations, and series Mensuration and geometry Elementary statistics and probability.

Analytical Aptitude

Logic: deduction and induction, Analogy, Numerical relations and reasoning. Spatial Aptitude Transformation of shapes: translation, rotation, scaling, mirroring, assembling, and grouping paper folding, cutting, and patterns in 2 and 3 dimensions.

Text Books:

1. "A Course in Phonetics and Spoken English" – T. Balasubramanian
2. "Effective Technical Communication" – M. Ashraf Rizvi
3. "Quantitative Aptitude for Competitive Examinations" – R.S. Aggarwal

Reference Books

1. "High School English Grammar & Composition" – Wren & Martin
2. "How to Speak, How to Listen" – Mortimer J. Adler
3. "Logical and Analytical Reasoning" – A.K. Gupta

NPTEL Course

1. Mastering Speaking and Presentations: A case Based Approach by Prof. Seema Singh, IIT Kharagpur
https://onlinecourses.nptel.ac.in/noc25_hs96/preview

Public Speaking And Aptitude (Lab)

Session 1: Vocabulary & Word Accent Mastery

- Activities:
  - Pre-Assessment: Conduct a quick spoken test to evaluate pronunciation and fluency
  - Vocabulary exercises using flashcards & interactive word-building games.
  - Accent exercises: Identify stress in disyllabic and polysyllabic words.

Session 2: Intonation & Rhythm in Speech

- Activities:
  - Introduction to rising and falling intonation with examples.
  - Roleplay exercises for practicing intonation in different scenarios (expressing surprise, asking questions, etc.).
  - Rhythm practice: Reading passages with proper pauses and stress patterns.

Session 3: Telephone Skills & Professional Conversations

- Activities:
  - Practicing making and taking calls with simulated dialogues.
  - Focus on voice modulation, clarity, and polite expressions.
  - Avoiding counterproductive questions in conversations – roleplay exercises.

Session 4: Body Language & Stage Presence

- Activities:
  - Mirror exercises to improve facial expressions and gestures.
  - Practicing posture, movement, and eye contact while speaking.
  - Group feedback and analysis using video recordings.

Session 5: Voice Modulation & Power of Pause

- Activities:
  - Exercises on voice projection and eliminating fillers.
  - Practicing pauses strategically to enhance speech impact.
  - Repetition and emphasis techniques using speech excerpts.

Session 6: Empathy & Heartfelt Communication

- Activities:
  - Interactive storytelling to practice emotional connection.
  - Exercises on active listening and empathetic responses.
  - Speech practice: delivering a short talk with an emotional appeal.

Session 7: Verbal Aptitude & Grammar Enhancement

- Activities:
  - Grammar quizzes and error detection in sentences.
  - Reading comprehension exercises with time constraints
  - Sentence structuring and narrative sequencing practice.

Session 8: Quantitative Aptitude – Data Interpretation & Computation

- Activities:
  - Solving numerical problems based on bar graphs, pie charts, and tables.
  - Quick estimation exercises using ratios, percentages, and logarithms.
  - Group challenges on permutations and combinations.

Session 9: Analytical Aptitude – Logical & Numerical Reasoning

- Activities:
  - Deduction and induction puzzles.
  - Solving analogy-based reasoning questions.
  - Speed tests for numerical relations and reasoning.

Session 10: Spatial Aptitude – Shape & Pattern Recognition

- Activities:
  - Hands-on paper folding and cutting exercises.
  - Visualization tasks for rotation, scaling, and mirroring of shapes.
  - Solving pattern-based problems in 2D and 3D space.

Semester VIII

CO23411 : Deep Learning

Teaching Scheme:	Credit: 04	Examination Scheme:
TH: 03 Hrs/Week PR: 02 Hrs/Week	TH Credit :03 PR Credit :01	Course Activity: 10 Mark In Semester: 30 Mark End Semester: 60 Mark Practical: 30 Mark

Prerequisite: Students are expected to have a good understanding of Discrete Mathematics, Data Structures and Algorithms

Course Objective:

- To introduce the fundamental concepts of neural networks and deep learning.
- To understand training algorithms and optimization techniques used in deep learning.
- To study different deep learning architectures for image and sequence data.
- To implement deep learning models using modern frameworks.
- To analyze the performance and applications of deep learning models.

Course Outcomes:

1. Explain the fundamentals of artificial neural networks and deep learning models.
2. Apply optimization techniques and regularization methods in neural networks.
3. Implement convolutional neural networks for image processing tasks.
4. Apply recurrent neural networks for sequence and time-series data.
5. Understand transformer-based models and modern NLP techniques.
6. Apply deep learning models to solve real-world applications.

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Mini Project
2. Survey on uses of deep learning with emerging technology presentation
3. Industry Visit
4. Seminar
5. Research Paper in deep learning domain

Course Contents

Mapping of Course Outcomes for Unit I		CO1
UNIT I	**Introduction to Deep Learning**	**07 Hours**

Introduction to Artificial Intelligence, Machine Learning and Deep Learning; limitations of traditional machine learning approaches; bias–variance tradeoff; overfitting and underfitting; motivation and advantages of deep learning.

Biological neuron vs artificial neuron; perceptron model; limitations of perceptron; implementation of logical functions (AND, OR, XOR); introduction to multilayer perceptron (MLP); architecture of feedforward neural networks. overview of deep learning frameworks such as TensorFlow and PyTorch;

Mapping of Course Outcomes for Unit II	CO2

UNIT II	Neural Network Training and Optimization	07 Hours

Activation functions including Sigmoid, Tanh, ReLU and Softmax; loss functions for regression and classification; gradient descent optimization; stochastic gradient descent and mini-batch learning; backpropagation algorithm.

Vanishing and exploding gradient problems; weight initialization techniques; optimization algorithms including Momentum, RMSProp and Adam; regularization methods such as L1/L2 regularization, dropout and early stopping.

Mapping of Course Outcomes for Unit III	CO3

UNIT III	Convolutional Neural Networks (CNN)	07 Hours

Convolution operation; padding and stride; pooling layers (max pooling, average pooling); architecture of CNNs; classic models such as LeNet, AlexNet and VGG; image classification using CNN; data augmentation techniques; introduction to transfer learning.

Mapping of Course Outcomes for Unit IV	CO4

UNIT IV	Recurrent Neural Networks (RNN)	06 Hours

Sequential data modeling; basic recurrent neural networks; backpropagation through time; vanishing gradient problem in RNNs; Long Short-Term Memory (LSTM); Gated Recurrent Unit (GRU); sequence prediction tasks; applications in language modeling and time-series forecasting.

Mapping of Course Outcomes for Unit V	CO5

UNIT V	Transformer Model and Modern NLP	06 Hours

Limitations of RNN models; attention mechanism; self-attention concept; introduction to Transformer architecture; positional encoding; overview of pretrained language models such as BERT and GPT; fine-tuning pretrained models for downstream NLP tasks.

Mapping of Course Outcomes for Unit VI	CO6

UNIT VI	Applications and Emerging Trends	06 Hours

Applications of deep learning in computer vision including image classification, object detection and image segmentation; applications in natural language processing such as machine translation, sentiment analysis and text summarization; recommendation systems

Introduction to transfer learning and fine-tuning; generative models such as autoencoders and Generative Adversarial Networks (GAN); introduction to transformer models and attention mechanisms; basic concepts of model deployment and inference; ethical issues, bias and challenges in deep learning systems.

Books and Other Resources

Text Books:

1. Goodfellow, I., Bengio, Y., Courville, A. – Deep Learning, MIT Press
2. **Aggarwal, Charu C.** *Neural Networks and Deep Learning: A Textbook*, Springer, 2018. **Publisher:** Springer **ISBN-13:** 978-3319944630
3. **Chollet, François.** *Deep Learning with Python*, 2nd Edition, Manning Publications, 2021.

Publisher: Manning Publications **ISBN-13:** 978-1617296864

Reference Books:

1. **Zhang, Aston; Lipton, Zachary C.; Li, Mu; Smola, Alexander J.** *Dive into Deep Learning*, Cambridge University Press, 2023 (Print Edition). **Publisher:** Cambridge University Press **ISBN-13:** 978-1108470049

2. **Géron, Aurélien.** *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, 2nd Edition, O'Reilly Media, 2019. **Publisher:** O'Reilly Media **ISBN-13:** 978-1492032649

Guidelines for Term Work Assessment :

Term work assessment will be based on overall performance of Laboratory assignments performed by a students. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, efficient codes, and punctuality.

Guidelines for Practical Examination :

Problem statements will be formed based on assignments and performance will be evaluated by Internal and External Examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation..

Guidelines for Laboratory Conduction :

Use of open source software is encouraged. Based on the concepts learned.
Operating System recommended :- 64-bit Open source Linux or its derivative Programming
Tools recommended: - Python, TensorFlow, PyTorch

Practical Assignments

1. Classify flower species into **three categories (Setosa, Versicolor, Virginica)** using input features such as sepal length, sepal width, petal length, and petal width. Implement a multilayer perceptron model and evaluate classification accuracy. Use **Iris Dataset** from Link: https://archive.ics.uci.edu/ml/datasets/iris

2. Image classification using deep neural networks. Example: Build a model to recognize **handwritten digits (0–9)** from grayscale images. Train and evaluate the model using fully connected neural networks and compare performance with CNN-based approaches. Use **MNIST Dataset** from Link: https://www.kaggle.com/datasets/hojjatk/mnist-dataset

3. Use a pretrained convolutional neural network model such as **VGG16 or ResNet** and fine-tune it to classify images into different categories such as animals or objects. The objective is to understand how pretrained models can be reused for new tasks with limited data. Use **Intel Image Classification Dataset** from Link: https://www.kaggle.com/datasets/puneet6060/intel-image-classification

4. Classify movie reviews into **positive reviews and negative reviews** based on the text content using recurrent neural networks such as LSTM. Perform text preprocessing and evaluate classification performance. Use **IMDB Movie Reviews Dataset** from Link: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

5. Predict **future airline passenger numbers** based on historical data using LSTM models. Visualize predictions and compare predicted values with actual values. Use **Airline Passenger Dataset** from Link: https://www.kaggle.com/datasets/rakannimer/air-passengers

6. Develop a model that predicts **movie ratings for users** based on previous rating patterns and recommend movies accordingly. Use **MovieLens Dataset** from Link: https://grouplens.org/datasets/movielens/

7. Generate descriptive captions for images using CNN for feature extraction and RNN/LSTM for text generation. Use **Flickr8k Dataset** from Link: https://www.kaggle.com/datasets/adityajn105/flickr8k

8. Classify **SMS messages into spam or ham (not spam)** using deep learning models. Perform text preprocessing, feature extraction and train a neural network classifier to detect unwanted messages. Use **SMS Spam Collection Dataset** from Link: https://archive.ics.uci.edu/ml/datasets/sms+spam+collection

9. Predict **human activities such as walking, sitting, standing and climbing stairs** using smartphone sensor data. Implement a neural network model to classify activities based on accelerometer and gyroscope signals. Use **Human Activity Recognition Dataset** from Link: https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

CO23412A : Natural Language Processing

Teaching Scheme:	Credit: 04	Examination Scheme:
TH: 03 Hrs/Week PR: 02 Hrs/Week	TH Credit :03 PR Credit :01	Course Activity: 10 Mark In Semester: 30 Mark End Semester: 60 Mark Oral: 30 Mark

Prerequisite: Students are expected to have a good understanding of Discrete Mathematics, Theory of Computation, Data Science and Big Data Analytics.

Course Objective:

- To be familiar with fundamental concepts and techniques of natural language processing (NLP)
- To acquire the knowledge of various morphological, syntactic, and semantic NLP tasks
- To develop the various language modeling techniques for NLP
- To use appropriate tools and techniques for processing natural languages
- To comprehend the advance real world applications in NLP domain.
- To Describe Applications of NLP and Machine Translations.

Course Outcomes:

On completion of the course, student will be able to–

CO1: Describe the fundamental concepts of NLP, challenges and issues in NLP

CO2: Analyze Natural languages morphologically, syntactical and semantically OR Describe the concepts of morphology, syntax, semantics of natural language

CO3: Illustrate various language modelling techniques

CO4: Integrate the NLP techniques for the information retrieval task

CO5: Demonstrate the use of NLP tools and techniques for text-based processing of natural languages

CO6: Develop real world NLP applications

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. NLP Mini Project
2. Survey on different NLP tools and Libraries.
3. Case Study on current NLP-related Real-World problem solutions
4. Seminar
5. Review of Research Paper in NLP domain

Mapping of Course Outcomes for Unit I	CO1

UNIT I	Introduction to Natural Language Processing	07 Hours

Introduction: Natural Language Processing, Why NLP is hard? Programming languages Vs Natural Languages, Are natural languages regular? Finite automata for NLP, Stages of NLP, Challenges and Issues(Open Problems) in NLP **Basics of text processing:** Tokenization, Stemming, Lemmatization, Part of Speech Tagging

Mapping of Course Outcomes for Unit II		CO2
UNIT II	**Language Syntax and Semantics**	**07 Hours**
Morphological Analysis: What is Morphology? Types of Morphemes, Inflectional morphology &Derivational morphology, Morphological parsing with Finite State Transducers (FST) **Syntactic Analysis**: Syntactic Representations of Natural Language, Parsing Algorithms, Probabilistic context-free grammars, and Statistical parsing **Semantic Analysis**: Lexical Semantic, Relations among lexemes & their senses – Homonymy, Polysemy, Synonymy, Hyponymy, WordNet, Word Sense Disambiguation (WSD), Dictionary based approach, Latent Semantic Analysis		
Mapping of Course Outcomes for Unit III		**CO3**
UNIT III	**Language Modelling**	**07 Hours**
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 **Word Embeddings/ Vector Semantics**: Bag-of-words, TFIDF, word2vec, doc2vec, Contextualized representations (BERT) **Topic Modelling**: Latent Dirichlet Allocation (LDA), Latent Semantic Analysis, Non Negative Matrix Factorization		
Mapping of Course Outcomes for Unit IV		**CO4**
UNIT IV	**Information Retrieval using NLP**	**06 Hours**
Information Retrieval: Introduction, Vector Space Model **Named Entity Recognition**: NER System Building Process, Evaluating NER System Entity Extraction, Relation Extraction, Reference Resolution, Coreference resolution, Cross Lingual Information Retrieval		
Mapping of Course Outcomes for Unit V		**CO5**
UNIT V	**NLP Tools and Techniques**	**06 Hours**
Prominent NLP Libraries: Natural Language Tool Kit (NLTK), spaCy, TextBlob, Gensim etc. **Linguistic Resources**: Lexical Knowledge Networks, WordNets, Indian Language WordNet (IndoWordnet), VerbNets, PropBank, Treebanks, Universal Dependency Treebanks **Word Sense Disambiguation**: Lesk Algorithm Walker's algorithm, WordNets for Word Sense Disambiguation.		
Mapping of Course Outcomes for Unit VI		**CO6**
UNIT VI	**Applications of NLP**	**06 Hours**
Machine Translation: Rule based techniques, Statistical Machine Translation (SMT), Cross Lingual Translation Sentiment Analysis, Question Answering, Text Entailment, Discourse Processing, Dialog and Conversational Agents, Natural Language Generation		
Books and Other Resources		

Text Books:

1. Jurafsky, David, and James H. Martin, ―Speech and Language Processing: An Introduction to Natural Language Processing‖, Computational Linguistics and Speech Recognition‖, ,PEARSON Publication
2. Manning, Christopher D., and nrich Schütze , ―Foundations of Statistical Natural Language Processing‖, Cambridge, MA: MIT Press

Reference Books:

1. Steven Bird, Ewan Klein, Edward Loper, ―Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit‖, O'Reilly Publication
2. Dipanjan Sarkar , ―Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data‖, Apress Publication ISBN: 9781484223871
3. Alexander Clark, Chris Fox, and Shalom Lappin, ―The Handbook of Computational Linguistics and Natural Language Processing‖, Wiley Blackwell Publications
4. Jacob Eisenstein, ―Natural Language Processing‖, MIT Press
5. Jacob Eisenstein, ―An Introduction to Information Retrieval‖, Cambridge University Press

e-Books :

1. https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
2. https://www3.cs.stonybrook.edu/~cse521/L16NLP.pdf

NPTEL Courses links:

1. https://nptel.ac.in/courses/106101007
2. https://nptel.ac.in/courses/106106211
3. https://nptel.ac.in/courses/106105158

Guidelines for Oral Examination :

The NLP practical oral examination expects evaluation based on student's understanding of NLP concepts, practical experiments, and the ability to explain code, outputs, and applications. Students must present their journal and mini project (if any) and answer questions on preprocessing, language models, tools, and NLP techniques. They may be asked to demonstrate or interpret their program. Evaluation focuses on conceptual clarity, implementation knowledge, problem-solving skills, and communication. Examiners ensure fairness by asking both conceptual and application-oriented questions.

Practical Assignments

1. Perform tokenization (Whitespace, Punctuation-based, Treebank, Tweet, MWE) using NLTK library. Use porter stemmer and snowball stemmer for stemming. Use any technique for lemmatization. Input / Dataset –use any sample sentence

2. Perform bag-of-words approach (count occurrence, normalized count occurrence), TF-IDF on data. Create embeddings using Word2Vec. Dataset to be used: https://www.kaggle.com/datasets/CooperUnion/cardataset

3. Perform text cleaning, perform lemmatization (any method), remove stop words (any method), label encoding. Create representations using TF-IDF. Save outputs. Dataset: https://github.com/PICT-NLP/BE-NLP- Elective/blob/main/3- Preprocessing/News_dataset.pickle

4. Create a transformer from scratch using the Pytorch library
   Morphology is the study of the way words are built up from smaller meaning bearing units. Study and understand the concepts of morphology by the use of add delete table.

5. Create a POS tagging visualizer to analyze and display grammatical tags in user-provided sentences.

6. **Mini Projects**: (Course coordinator may ask students to implement any 2 mini projects from the below list)
   a. Build a sentiment analysis system to classify user reviews or tweets as positive, negative, or neutral.
   b. Develop a fake news detection model to classify news articles as real or fake.
   c. Build a resume parser to extract key information such as skills, education, and experience from resumes.
   d. Develop a mini machine translation system to translate text between English and Indian languages
   e. Create an automatic question generator that forms simple questions from a given passage.
   f. Build an emotion detection model to classify text into emotions like joy, anger, or sadness.
   g. Develop a mini machine translation system to translate text between English and Indian languages.

7. Implement a plagiarism checker to detect similarity between documents using NLP techniques.

CO23412B: Soft Computing

Teaching Scheme:	Credit: 04	Examination Scheme:
TH: 03 Hrs/Week PR: 02 Hr/Week	TH Credit :03 PR Credit :01	Course Activity: 10 Mark In Semester: 30 Mark End Semester: 60 Mark Oral: 30 Mark

Prerequisite:

- Basic knowledge of **probability**, **statistics**, and **linear algebra**
- Fundamentals of **Artificial Intelligence and Machine Learning**
- Basic programming skills in **Python**
- Understanding of algorithms, optimization, and data structures

Course Objective:

- Introduce principles of Soft Computing and explain how it differs from hard computing.
- Develop understanding of **Fuzzy Logic**, **Neural Networks**, and **Evolutionary Computation**.
- Provide ability to design real-life problem-solving systems using soft computing techniques.
- Train students to apply hybrid models for optimization and decision-making.
- Enable students to implement soft-computing algorithms through practical assignments.

Course Outcomes:

CO1: Explain concepts of soft computing, its importance, and applications.
CO2: Construct and analyze **fuzzy logic systems**, fuzzy inference, fuzzification and defuzzification.
CO3: Design, train, and evaluate **Artificial Neural Networks** for classification/regression applications.
CO4: Apply **genetic algorithms** and evolutionary techniques for solving real-world optimization problems.
CO5: Develop hybrid soft-computing models integrating ANN, Fuzzy, and GA for intelligent systems.
CO6: Implement soft-computing algorithms in Python/MATLAB and demonstrate solutions through mini projects.

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Mini assignment based on Unit completion
2. Seminar on soft computing applications
3. Review of one IEEE paper
4. Practical performance & viva
5. Class participation

Course Contents

Mapping of Course Outcomes for Unit I	CO1

UNIT I	Introduction to Soft Computing	07 Hours

Hard vs Soft Computing, Need for Soft Computing, Components: Fuzzy Logic, Neural Networks, Evolutionary Algorithms, Applications in engineering, medicine, robotics, prediction systems, Soft computing vs traditional AI

Mapping of Course Outcomes for Unit II	CO2

UNIT II	Fuzzy Logic and Fuzzy Systems	07 Hours
Fuzzy sets, membership functions, Operations on fuzzy sets, Fuzzy rules and fuzzy reasoning, Fuzzy inference systems (Mamdani, Sugeno), Fuzzification and Defuzzification techniques, Fuzzy decision-making and control, Applications in control systems and expert systems		
Mapping of Course Outcomes for Unit III	**CO3**	
UNIT III	**Artificial Neural Networks (ANN)**	**07 Hours**
Biological neuron model and artificial neuron, Types of neural networks, Perceptron and multilayer perceptron, Activation functions, Backpropagation algorithm, ANN training, testing, validation, Case studies: pattern recognition / prediction		
Mapping of Course Outcomes for Unit IV	**CO4**	
UNIT IV	**Advanced Neural Networks**	**06 Hours**
Radial Basis Function Networks (RBFN), Self-Organizing Maps (SOM) / Kohonen networks, Recurrent Neural Networks (RNN) – Basics, Deep Learning overview, Applications of ANN		
Mapping of Course Outcomes for Unit V	**CO5**	
UNIT V	**Evolutionary Algorithms & Genetic Algorithms**	**06 Hours**
Need for evolutionary computing, GA operators: selection, crossover, mutation, fitness function, GA flow and convergence, Particle Swarm Optimization (PSO) – fundamentals, Comparison of GA, PSO, DE, Applications: optimization & search problems		
Mapping of Course Outcomes for Unit VI	**CO6**	
UNIT VI	**Hybrid Systems & Case Studies**	**06 Hours**
Neuro-Fuzzy systems, Fuzzy-GA hybrid, ANN-GA hybrid, Adaptive Neuro-Fuzzy Inference System (ANFIS), Real-life case studies (at least two):Medical diagnosis, Traffic prediction, Stock market prediction, Image recognition, Future trends in soft computing		

Books and Other Resources

Text Books:

1. **S. Rajasekaran, G. A. Vijayalakshmi Pai**, *Neural Networks, Fuzzy Logic and Genetic Algorithms*, PHI.
2. **Jang, Sun, Mizutani**, *Neuro-Fuzzy and Soft Computing*, Pearson.

Reference Books:

1. **Simon Haykin**, *Neural Networks and Learning Machines*, Pearson.
2. **Goldberg**, *Genetic Algorithms in Search, Optimization and Machine Learning*.
3. **Zimmermann**, *Fuzzy Set Theory and Its Applications*.
4. **Eberhart & Kennedy**, *Particle Swarm Optimization*.

Guidelines for Practical Examination :

Problem statements will be formed based on assignments and performance will be evaluated by Internal and External Examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student‘s understanding of the fundamentals, effective and efficient implementation..

Guidelines for Laboratory Conduction :

Weekly practical sessions based on theory units. Use Python libraries: NumPy, SciPy, scikit-fuzzy, TensorFlow/Keras (optional). Every student should maintain a journal with: Aim, theory, algorithm, code, output, conclusion. Conduct at least 1 small-scale mini-project (e.g., classifier, prediction system).

Practical Assignments (Any 10)

1. Exploring Git Commands through Collaborative Coding.
2. Design and Analysis of DevOps Lifecycle for a Web Application
3. Demonstrate Container Orchestration using Kubernets.
4. Applying CI/CD Principles to Web Development Using Jenkins, Git, and Local HTTP Server
5. Exploring Containerization and Application Deployment with Docker
6. Create a simple project, push it to a remote repository on GitHub, and create a new branch. Merge this branch into the main branch and display a chronological history of commits.
7. Create Ansible playbook for a simple web application infrastructure.
8. Install Docker on your system and create a simple "Hello, World!" application using HTML.Create a Dockerfile to containerize the application, using an official web server image as the base. Build the Docker image, tag it, and run a container, making the application accessible on a local port (e.g., http://localhost:8080).
9. To Setup and Run Selenium Tests in Jenkins Using Maven.
10. To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.
11. To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet.
12. Implement basic monitoring and logging mechanisms for a simple web application.

CO23412C : Software Architecture and Design Patterns

Teaching Scheme:	Credit: 05	Examination Scheme:
TH: 03 Hrs/Week	TH Credit :03	Course Activity: 10 Marks
PR: 02 Hrs/Week	PR Credit :02	In Semester: 30 Marks
		End Semester: 60 Marks
		Oral: 30 Marks

Prerequisite: Students are expected to familiar with Object Oriented Programming, Software Engineering

Course Objective:

- To introduce students to the basic concepts and techniques of SADP.
- To acquaint with the interaction between quality attributes and software architecture
- To understand software architecture with case studies and explore with examples
- To understand different architectural designs, transform them into proper model and document them

Course Outcomes:

CO1: Design and analyze an application using UML modeling as fundamental tool

CO2: Evaluate software architectures

CO3: Use appropriate architectural styles and software design patterns

CO4: Apply appropriate modern tool for designing and modeling

Course Activity:

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Group Discussion
2. Poster Presentation
3. Seminar
4. Quiz

Course Contents

Mapping of Course Outcomes for Unit I	CO1

UNIT I	Introduction	07 Hours

Definition of Software Modelling, Principles of modelling. Evolution of Software Modeling and Design Methods: Object oriented analysis and design methods, Concurrent, Distributed Design Methods and Real-Time Design Methods, Model Driven Architecture (MDA), 4+1 Architecture, Overview of Unified Modeling Language (UML) and its diagram types, Process Unified Process / Rational Unified Process inception, elaboration, construction, transition How various components fit in the life cycle

Mapping of Course Outcomes for Unit II	CO2

UNIT II	Software Architecture	08 Hours

Definition of Software Architecture and its use cases, Importance of Software Architecture, Architectural structures and views, Architectural Styles: Pipes and Filters, Data Abstraction and Object – Oriented Organization, Event-Based, Implicit Invocation, Layered Systems, Repositories, Interpreters, Other familiar Architectures, Heterogeneous

Architectures. Quality Attributes: Architecture and Requirements, Quality Attributes and Considerations

Mapping of Course Outcomes for Unit III		CO3
UNIT III	**Introduction to Patterns**	**07 Hours**
Meaning of Pattern, Definition of Design Pattern, What makes a Pattern (GOF), Describing Design Patterns, Pattern Categories & Relationships between Patterns, Organizing the Catalogue, Patterns and Software Architecture		
Mapping of Course Outcomes for Unit IV		**CO3**
UNIT IV	**Design Patterns**	**06 Hours**
Design Patterns: Introduction, Different approaches to select Design Patterns. Creational patterns: Singleton, Factory, Structural pattern: Adapter, Decorator, Proxy. Behavioral Patterns: Iterator, Observer Pattern with applications..		
Mapping of Course Outcomes for Unit V		**CO3**
UNIT V	**GRASP(General Responsibility Assignment Software Patterns)**	**06 Hours**
Expert, Creator, High Cohesion, Low Coupling, Controller, Polymorphism, Pure Fabrication, Indirection, Don't Talk to Strangers		
Mapping of Course Outcomes for Unit VI		**CO4**
UNIT VI	**Architectural Design and Documentation**	**06 Hours**
Architecture in the Life Cycle: Architecture in Agile Projects, Architecture and Requirements, Designing an Architecture. Documenting Software Architecture: Notations, Choosing and Combining views, Building the documentation Package, Documenting Behavior, Documenting Architecture in an Agile Development Project.		

Books and Other Resources

Text Books:

1. Jim Arlow, Ila Neustadt, "UML 2 and the unified process –practical object-oriented analysis and design", Addison Wesley, Second edition, ISBN 978-0201770605.
2. Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", Second Edition, Pearson ,ISBN 978-81-775-8996-2 Curriculum for Third Year of Computer Engineering (2019 Course), Savitribai Phule Pune University http://collegecirculars.unipune.ac.in/sites/documents/Syllabus2020/Forms/AllItems.aspx #76/87
3. Erich Gamma, "Design Patterns", Pearson, ISBN 0-201-63361-2Design Patterns – Elements of Reusable Object-oriented Software By E. Gamma, Richard Helm, Ralph Johnson , John Vlissides (GoF)

Reference Books:

1. Design Pattern – Oriented Software Architecture (POSA) Volume 1. By : Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal.
2. Software Architecture in Practice. By Len Bass, Paul Clements, Rick Kazman
3. Applying UML and Patterns By Craig Larman.
4. Software Architecture- Perspectives on an emerging discipline by Mary shaw and David Garlan
5. Head First Design Pattern by Kathy Sierra, Bert Bates, Elisabeth Robson, Eric Freeman Publisher: OReilly

Media, Inc. Building Microservices-Designing Fine-Grained Systems By Sam Newman Publisher: OReilly Media

7. Design patterns in Java by Douglas Schmidt Publisher O'Reilly SPPU M.Sc. Computer Science Part-II Syllabus 2024-25 8
8. Professional Java Development with the Spring Framework 1st Edition by Rod Johnson, Alef Arendsen, Thomas Risberg, Colin Sampaleanu ; WROX publication
9. Mastering Spring 5: An effective guide to build enterprise applications using Java Spring and Spring Boot framework, 2nd Edition by Ranga Rao Karanam ; PACKT publishing

The Unified Modeling Language reference manual Second Edition By James Rumbaugh, Ivar Jacobson, Grady Booch Publisher: Pearson Higher Education

Guidelines for Term Work Assessment:

Term work assessment will be based on overall performance of Laboratory assignments performed by a students. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, efficient codes, and punctuality.

Guidelines for Laboratory Conduction:

Use of open source software is encouraged. Based on the concepts learned.

Operating System recommended:- 64-bit Open source Linux or its derivative Programming

Practical Assignments

1. Design Structural UML Diagrams for any suitable system.
2. Design Behavioral UML Diagrams for any suitable system.
3. Identify classes in project and apply Expert & Creator patterns for responsibility distribution.
4. Implement I/O Decorator for converting uppercase letters to lower case letters.
5. Design Singleton pattern for any suitable system.
6. Design Adapter pattern for any suitable system.
7. **Mini Project:** Apply design pattern for any suitable system like online library management system

CO23413A : DevOps Fundamentals

Teaching Scheme	Credit: 03	Examination Scheme
Teaching Scheme: **TH: 02 Hrs/Week** **PR: 02 Hrs/Week**	**TH Credit :02** **PR Credit :01**	**End Semester: 60 Mark** **Course Activity: 10 Mark** **Oral: 30 Mark**

Prerequisite Courses: Software Engineering and Project Management, Cloud Computing

Companion Course, if any:

Course Objectives:

1. To understand the need of DevOps as a software engineering practice.
2. To understand the background of DevOps Evolution.
3. To know and understand the concept of Continuous Integration Continuous Delivery (CICD).
4. To learn the concept of continuous deployment and test strategies.
5. To learn the monitoring system and reliability engineering.
6. To explore the emerging tools used in the DevOps lifecycle.

Course Outcomes:

On completion of the course, students will be able –

CO1. Understand the fundamental concepts of DevOps
CO2. Link the background of DevOps with other technologies
CO3. Comprehend the concept of continuous integration and continuous delivery
CO4. Compare various stages of continuous deployment and test strategies
CO5. Justify the importance of monitoring system and reliability engineering
CO6. Use the latest tools in DevOps

Course Activity:

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Mini Project
2. Case Study
3. Industry Visit
4. Seminar

COURSE CONTENTS

Mapping of Course Outcomes for Unit I		CO1
Unit I	**Introduction to DevOps and the Culture**	**(6 hrs)**

What is DevOps? Role of DevOps Engineer, Developer responsibility, Introduction to Continuous Integration and Continuous Delivery Policies DevOps Culture: Integration of development, operations, and automation practices in IT organizations. Process automation, Agile Practices, Reason for adopting DevOps, What and Who Are Involved in DevOps? Changing the Coordination, Introduction to DevOps pipeline phases, Defining the Development Pipeline, Centralizing the Building Server, Monitoring Best
Practices, Best Practices for Operations.

Mapping of Course Outcomes for Unit II		CO2
Unit II	**Microservices Architecture and Cloud Native Development**	**(6 hrs)**
Monolithic applications, Introduction to microservice architecture, implementing a microservices Architecture, Pros and Cons of a microservice Architecture, Characteristics of microservice architecture, Monolithic applications and microservices compared, microservices best practices, Deployment strategies, Introduction to cloud computing, cloud computing deployment models, service models, why to use cloud, Principles of container-based application design, Introduction to Docker, Serverless computing, orchestration, Difference between orchestration and automation.		
Mapping of Course Outcomes for Unit III		**CO3**
Unit III	**Continuous Integration and Test-Driven Development**	**(6 hrs)**
Introduction to continuous integration, time to market and quality, Build in a Continuous Integration Scenario, Code Repository Server, Continuous Integration Server, Introduction to Continuous Delivery and chain, Differentiate Continuous Integration and Continuous Delivery, Strategies for Continuous Delivery, Benefits of Continuous Integration and Continuous Delivery, Designing a CI and CD System, Building Continuous Integration and Continuous Delivery Pipelines, Continuous Database Integration, Preparing the Build for Release, Identifying the Code in the Repository, Creating Build Reports, Putting the Build in a Shared Location, Releasing the Build		
Mapping of Course Outcomes for Unit IV		**CO4**
Unit IV	**Continuous Deployment and Orchestration**	**(6 hrs)**
Implementing a testing Strategy: Types of Tests, Integration testing, managing defect backlogs, what is Continuous Deployment? Changes moving through the deployment pipeline, Trade-offs in the deployment pipeline, Basic Deployment pipeline, Deployment pipeline practices & Commit stage, Automated Acceptance Test Gate, Subsequent test stages, preparing to release, Implementing a deployment pipeline		
Mapping of Course Outcomes for Unit V		**CO5**
Unit V	**Continuous Monitoring and Site Reliability**	**(6 hrs)**
What is a monitoring system? Factors involved in monitoring systems, why monitoring is important, white-box and black-box monitoring, building a monitoring system, monitoring infrastructure and applications, collecting data, logging, creating dashboard, behavior driven monitoring, what is site reliability engineering? SRE and DevOps, roles, and responsibilities of SRE, common tools used by SREs		
Mapping of Course Outcomes for Unit VI		**CO6**
Unit VI	**DevOps Tooling and Case Studies**	**(6 hrs)**
Continuous Development/ Version Control: Git, Serverless orchestration: Kubernetes, Container Technology: Docker, Continuous Integration: Jenkins, Continuous delivery: Jenkins, Continuous Deployment: Ansible, Puppet Continuous Testing: Selenium, Monitoring: Prometheus, Bug tracking tool: Jira,elk stack. Case study: Spotify: Using Docker, Bank of New Zealand, EtSy.		
Textbooks:		
1. PierluigiRiti, “Pro DevOps with Google Cloud Platform”, Apress, ISBN: 978-1-4842-3896-7.		
2. Katrina Clokie, “A Practical Guide to Testing in DevOps”, Lean Publishing published on 2017-08-01		

3. Jez Humble and David Farley, "Continuous Delivery", Pearson Education, Inc, ISBN: 978–0–321–60191–9

Reference Books:

1. Viktor Farcic, "The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices"

2. Jennifer Davis and Katherine Daniels, "Effective DevOps: Building a Culture of Collaboration, Anity, and Tooling at Scale", O'Reilly Media, Inc., ISBN: 978-1-491-92630-7

3. Sanjeev Sharma and Bernie Coyne, "DevOps for Dummies", John Wiley & Sons, Inc., 2nd IBM Limited Edition, ISBN: 978-1-119-04705-6

Web Links:

1. https://www.redhat.com/en/resources/cloud-native-container-design-whitepaper
2. https://www.redhat.com/en/topics/cloud-native-apps/what-is-serverless
3. https://www.redhat.com/en/topics/automation/what-is-orchestration
4. https://www.atlassian.com/continuous-delivery/continuous-integration
5. https://www.flagship.io/glossary/site-reliability-engineer/
6. https://docs.microsoft.com/en-us/learn/paths/intro-to-vc-git/
7. https://www.javatpoint.com/kubernetes
8. https://www.javatpoint.com/docker-tutorial
9. https://www.javatpoint.com/jenkins
10. https://www.javatpoint.com/jenkinss
11. https://www.javatpoint.com/ansible
12. https://www.javatpoint.com/selenium-tutorial
13. https://prometheus.io/docs/introduction/overview/
14. https://www.javatpoint.com/jira-tutorial
15. https://www.geeksforgeeks.org/what-is-elastic-stack-and-elasticsearch/

Practical Assignments

1. Exploring Git Commands through Collaborative Coding.
2. Demonstrate Container Orchestration using Kubernets.
3. Applying CI/CD Principles to Web Development Using Jenkins, Git, and Local HTTP Server
4. Exploring Containerization and Application Deployment with Docker
5. Create a simple project, push it to a remote repository on GitHub, and create a new branch. Merge this branch into the main branch and display a chronological history of commits.
6. Create Ansible playbook for a simple web application infrastructure.
7. Install Docker on your system and create a simple "Hello, World!" application using HTML.Create a Dockerfile to containerize the application, using an official web server image as the base. Build the Docker image, tag it, and run a container, making the application accessible on a local port (e.g., http://localhost:8080).
8. To Setup and Run Selenium Tests in Jenkins Using Maven.
9. To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.
10. To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet**.**

CO23413B : Blockchain Technology

Teaching Scheme:	Credit: 03	Examination Scheme:
TH: 03 Hrs/Week PR: 02 Hrs/Week	TH Credit :02 PR Credit :01	Course Activity: 10 Mark End Semester:60 Marks Oral: 30 Marks

Prerequisite: Students are expected to familiar with Object Oriented Programming, Software Engineering

Course Objective:–

- To understand the fundamentals of blockchain and distributed ledger technology.
- To learn cryptographic concepts essential for blockchain.
- To study various blockchain consensus mechanisms.
- To develop smart contracts and DApps using Ethereum.
- To explore real-world applications of blockchain across multiple domains.
- To understand blockchain challenges, security issues, and emerging trends.

Course Outcomes:

CO1: Explain blockchain architecture, features, and underlying cryptographic oncepts.

CO2: Compare and evaluate consensus algorithms for different blockchain systems.

CO3: Develop and deploy smart contracts and simple DApps on Ethereum.

CO4: Identify and apply blockchain solutions in real-world domains.

CO5: Use blockchain development tools and test networks effectively.

CO6 : Analyze blockchain challenges, security issues, and future technological trends.

Course Activity:

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Group Discussion
2. Poster Presentation
3. Seminar
4. Quiz

Course Contents

Mapping of Course Outcomes for Unit I	CO1

UNIT I	Introduction to Blockchain & Distributed Ledger Technology	07 Hours

Distributed systems, Centralized–Decentralized–Distributed architectures, Blockchain introduction, Blockchain features, Block structure, Hash and timestamp, Merkle tree, Types of blockchain, Blockchain vs database, Transaction lifecycle, Blockchain platforms overview

Mapping of Course Outcomes for Unit II	CO2

UNIT II	Cryptographic Foundations	07 Hours

Hash functions, SHA-256, Keccak-256, Public key cryptography, ECDSA, Digital signatures, Merkle trees, Nonce, Proof-of-Work fundamentals, Wallets and key management, Cryptographic security,

Blockchain attack types

Mapping of Course Outcomes for Unit III	CO3

UNIT III	Consensus Mechanisms	07 Hours

Consensus need, Proof-of-Work, Proof-of-Stake, Delegated PoS, Proof-of-Authority, PBFT, Mining and validators, Rewards and incentives, Forks (hard and soft), Consensus comparison

Mapping of Course Outcomes for Unit IV	CO4

UNIT IV	Ethereum & Smart Contracts Development	06 Hours

Ethereum architecture, EVM, Accounts and gas, Ethereum transactions, Smart contract basics, Solidity syntax, Variables and functions, Events and modifiers, Contract deployment, Remix IDE, Ganache, MetaMask, Web3.js/Ethers.js, Smart contract security issues

Mapping of Course Outcomes for Unit V	CO5

UNIT V	Blockchain Applications	06 Hours

Blockchain in finance, DeFi, Supply chain applications, Healthcare applications, Identity management, NFT concepts, ERC-20 token standard, ERC-721 standard, Use case studies (Bitcoin, Ethereum, Hyperledger)

Mapping of Course Outcomes for Unit VI	CO6

UNIT VI	Challenges, Security, and Future Trends	06 Hours

Scalability issues, Layer-2 solutions, Sidechains, Rollups, State channels, Privacy techniques, Zero-knowledge proofs, zk-SNARKs, Blockchain governance, Environmental impact, Smart contract vulnerabilities, Web3, DAO, CBDC, Quantum-safe blockchain

TEXTBOOKS

1. Mastering Blockchain by Imran Bashir, Packt Publishing, 3rd Edition (2020), ISBN: 978-1839213199
2. Mastering Ethereum by Andreas M. Antonopoulos & Gavin Wood, O'Reilly Media, 1st Edition (2018), ISBN: 978-1491971949
3. Blockchain Basics: A Non-Technical Introduction by Daniel Drescher, Apress, 1st Edition (2017), ISBN: 978-1484226032
4. Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions by Joseph J. Bambara & Paul R. Allen, McGraw-Hill, 1st Edition (2018), ISBN: 978- 1260115871

REFERENCE BOOKS

1. Bitcoin Whitepaper by Satoshi Nakamoto, Public Domain (2008), No ISBN/ISSN
2. Ethereum Yellow Paper by Gavin Wood, Ethereum Foundation, Technical Specification, No ISBN/ISSN
3. Building Blockchain Projects by Narayan Prusty, Packt Publishing, 2nd Edition (2017), ISBN: 978-1787122147
4. Blockchain Revolution by Don Tapscott & Alex Tapscott, Penguin/Portfolio, Updated Edition (2018), ISBN: 978-1101980149

5. Hands-On Smart Contract Development with Solidity and Ethereum by Kevin Solorio, Randall Kanna & David Hoover, O'Reilly Media, 1st Edition (2019), ISBN: 978- 1492045267

PRACTICALS (Minimum 10)

1. Installation and configuration of MetaMask, Ganache, Remix IDE.
2. Create wallet accounts, manage public/private keys, use test networks.
3. Implement SHA-256 hashing, Keccak hashing, and Merkle tree creation.
4. Write a simple "Hello Blockchain" smart contract in Solidity.
5. Smart contract with state variables and CRUD operations.
6. Develop an ERC-20 token smart contract.
7. Develop a basic ERC-721 NFT smart contract.
8. Build a simple DApp using Web3.js or ethers.js.
9. Deploy and verify a smart contract on a public testnet (Goerli/Polygon Mumbai).
10. Analyze blocks and transactions using Etherscan or block explorers.
11. Optional: Simulate PoW/PoS consensus in code.

Term Work Activity

Option A – Mini Project (Preferred)

Students (groups of 2–3) must develop a mini blockchain application, such as:

- Blockchain-based voting system
- Supply chain tracking system
- NFT marketplace prototype
- Certificate verification using blockchain
- Crypto wallet simulation

Marks Distribution:

- Documentation & Innovation – 4 Marks
- Project Implementation & Demo – 4 Marks
- Viva – 2 Marks

Option B – Research Review (Alternative)

Review of a recent blockchain research paper (last 5 years).

- Summary – 4 Marks
- Analytical Review – 4 Marks
- Presentation – 2 Marks

CO23413C: Quantum Computing

Teaching Scheme:	Credit: 03	Examination Scheme:
TH: 02 Hrs/Week	TH Credit :02	Course Activity: 10 Mark
PR: 02 Hrs/Week	PR Credit :01	End Semester: 60 Mark
		Oral : 30 Mark

Prerequisite: Students are expected to have a good understanding of Linear Algebra, Classical Computing

Course Objective:

- To provide basic understanding of quantum computing concepts, qubits, and key application areas.
- To introduce essential mathematical and quantum mechanics foundations for quantum computing.
- To introduce basic quantum computing architecture and qubit technologies.
- To introduce fundamental quantum gates and basic quantum circuit construction and simulation.
- To introduce students to the Qiskit framework for building and executing quantum circuits.
- To introduce fundamental quantum algorithms and the principles behind quantum speedup.

Course Outcomes:

1. Differentiate classical computing from quantum computing and outline key application areas of quantum technologies.
2. Describe key mathematical tools and quantum concepts such as complex vectors, unitary matrices, and state vectors.
3. Identify major qubit types and key hardware concepts.
4. Apply single-qubit and multi-qubit gates to build and simulate simple quantum circuits.
5. Create and run basic quantum circuits in Qiskit using IBM Quantum tools.
6. Explain basic quantum algorithms and their advantages over classical approaches.

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Quantum Mini Project
2. Survey on uses of Quantum Computing presentation
3. Industry Visit
4. Seminar
5. Research Paper in quantum computing domain

Course Contents

Mapping of Course Outcomes for Unit I	CO1	
UNIT I	**Foundation of Quantum Computing**	**07 Hours**

Introduction to Quantum Computing - Evolution of computing: Classical vs Quantum , Quantum advantage and limitations , Quantum Computing Basics, Difference between classical bits and quantum bits (qubits), Use of quantum computing, Quantum Computing Applications, Quantum States Overview of fields benefiting from quantum computing (cryptography, optimization, material science).

Mapping of Course Outcomes for Unit II	CO2	
UNIT II	**Background Mathematics and Physics**	**07 Hours**

Complex Numbers & Basics of Linear Algebra- Complex numbers, Euler's formula , Vectors and matrices , Inner product, norms ,Orthonormal basis, **Matrix Operations for Quantum Computing-** Matrix multiplication Hermitian, unitary matrices , Pauli matrices , Tensor product of vectors/matrices, **Introduction to Quantum Mechanics -**Postulates of quantum mechanics, State vectors and wave functions, Operators and observables.

Mapping of Course Outcomes for Unit III	**CO3**

UNIT III	**Building Blocks of Quantum Computers and Quantum Correlations**	**07 Hours**

Introduction to Quantum Computing Architecture- Classical vs quantum hardware, Quantum information processing, Logical qubits vs physical qubits**, Physical Implementations of Qubits-** Superconducting qubits, Trapped ion qubits, Photonic qubits, Spin-based and NV-centre qubits, **Key metrics**: coherence time, gate fidelity, scalability.

Mapping of Course Outcomes for Unit IV	**CO4**

UNIT IV	**Quantum Gates and Operations**	**06 Hours**

Single-Qubit Quantum Gates: Identity (I) Gate , Pauli Gates: X, Y, Z, Hadamard (H) Gate , Phase Gates: S (Phase), T ($\pi/8$), **Multi-Qubit Gates-** CNOT (CX) , Controlled-Z (CZ) , Toffoli (CCNOT), SWAP gate, **Quantum Circuit-** Building quantum circuits using gates**,** Representing circuits using QASM, Circuit simulation basics, Measurement gates.

Mapping of Course Outcomes for Unit V	**CO5**

UNIT V	**Qiskit Simulator**	**06 Hours**

Introduction to Qiskit- Overview of Qiskit framework , Installation and setup (pip, Jupyter), Working with IBM Quantum platform, **Qiskit Basics**-Qiskit workflow: Build → Execute → Analyze, Quantum circuits in Qiskit, Qubit creation & initialization, Applying basic gates (X, H, RX, CNOT), Visualization basics.

Mapping of Course Outcomes for Unit VI	**CO6**

UNIT VI	**Quantum Algorithms**	**06 Hours**

Introduction to Quantum Algorithms: Classical vs quantum algorithmic thinking, Oracle-based algorithms (black-box idea), Quantum Fourier Transform (QFT) – concept only, Amplitude amplification (intro), **Deutsch & Deutsch–Jozsa Algorithms-** Problem definition , Balanced vs constant functions, Circuit implementation, **Grover's Search Algorithm-** Unstructured search problem , Oracle and diffusion operator,
Amplitude amplification concept, Complexity: $O(\sqrt{N})$, Applications and limitations.

Books and Other Resources

Text Books:

1. Quantum Computation and Quantum Information by Michael Nielsen and Isaac Chuang, Publisher: Cambridge University Press.
2. Quantum Computing for Everyone by Chris Bernhardt.
3. Qiskit Textbook: A free, open-source online resource provided by IBM.

Reference Books:

1. Introduction to Quantum Computing by Phillip Kaye, Raymond Laflamme, and Michele Mosca.
2. Quantum Algorithms Via Linear Algebra: A Primer by Richard J. Lipton and Kenneth W. Regan

Guidelines for Term Work Assessment :

Term work assessment will be based on overall performance of Laboratory assignments performed by a students. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, efficient codes, and punctuality.

Guidelines for Practical Examination :

Problem statements will be formed based on assignments and performance will be evaluated by Internal and External Examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student‘s understanding of the fundamentals, effective and efficient implementation..

Guidelines for Laboratory Conduction :

Use of open source software is encouraged. Based on the concepts learned.
Operating System recommended :- 64-bit Open source Linux or its derivative Programming
Tools recommended: - Jupyter Notebook, Qiskit, IBM Quantum Platform.

Practical Assignments

1. Set up the Qiskit quantum-computing framework on a local system by installing the required Python environment, dependencies, and Qiskit packages.
2. Develop a foundational understanding of linear algebra concepts essential for quantum computing by exploring vector operations, vector multiplication, and the tensor product.
3. Implement the identity matrix for quantum systems of increasing size by constructing the 1-qubit, 2-qubit, and 3-qubit identity operators.
4. Implement the Pauli gates ,Y, and Z and analyze their effects on single-qubit quantum states.
5. Implement the Hadamard (H) gate and examine its effect on single-qubit states.
6. Implement fundamental two-qubit quantum gates and analyze their effects on multi-qubit states.
7. Implement key three-qubit quantum gates and evaluate their effects on multi-qubit states.
8. Design and implement a basic quantum circuit by arranging single-qubit and multi-qubit gates in a logical sequence.
9. Implement Circuit Formation-2 by constructing a multi-step quantum circuit that combines single-qubit gates, controlled operations, and sequential transformations.
10. Analyze the use of IBM Cloud for quantum computing by exploring its quantum hardware access, cloud-based execution model, and development tools such as Qiskit Runtime.

CO23414PR : Project		
Teaching Scheme: PR: 08 Hrs/Week	**Credit: 04** PR Credit :04	**Examination Scheme:** Term Work: 80 Mark Oral: 50 Mark

Prerequisite: Students are expected to have a good understanding of including problem definition, literature review, and initial design, Basic understanding of software engineering principles and the Software Development Life Cycle (SDLC), Knowledge of programming languages, tools, and technologies, Familiarity with UML diagrams, Ability to perform testing methodologies, Experience with documentation standards

Course Objective:

- To adhere carefully to the Software Development Life Cycle (SDLC) and achieve the goals of the proposed project.
- To carry out thorough and comprehensive testing before the system is deployed.
- To verify and confirm the outcomes of the work performed.
- To compile and present the completed work in a well-structured report.

Course Outcomes:

1. Demonstrate the ability to carry out independent research work.
2. Evaluate and interpret results with critical insight.
3. Organize, document, and present original findings clearly while positioning unresolved issues appropriately.
4. Connect relevant methods and findings from existing literature, current research, and potential future research directions.
5. Understand the real-world applications and limitations of the specialized topic.

Guidelines

In Project Work, the student is required to complete the remaining part of the project, which includes choosing appropriate technologies and tools, carrying out necessary installations, developing UML diagrams, performing testing, and presenting results.

The work should also involve discussing performance using data tables for each selected parameter, comparing the outcomes with existing or known algorithms/systems, and conducting a comparative analysis along with result validation and conclusions.

The student must then prepare and submit the project report in the prescribed standard format, duly approved by the project guide and the Head of the Department/Institute, to ensure successful completion of the project.