Lingaya's Vidyapeeth

Deemed-to-be-University u/s 3 of UGC Act 1956, Government of India
NAAC ACCREDITED
Approved by MHRD / AICTE / PCI / BCI / COA / NCTE
Nachauli, Jasana Road, Faridabad- 121002 (Haryana)
Website: www.lingayasvidyapeeth.edu.in | Ph: 0129-2598200-05

COURSE PLAN & COURSE DATA SHEET

PROGRAM: Btech	DEGREE: UG
COURSE: Software Engineering	SEMESTER: VI CREDITS: 3
COURSE CODE: CS - 304 REGULATION: NA	COURSE TYPE: CORE
COURSE AREA/DOMAIN: IT	CONTACT HOURS: 3+1 (Tutorial) hours/Week.
CORRESPONDING LAB COURSE CODE (IF ANY): NA	LAB COURSE NAME (IF ANY): NA

PROGRAM EDUCATIONAL OBJECTIVES:

- Graduates will be proficient in designing, implementing, and testing software systems using contemporary tools and practices.
- Graduates will demonstrate the ability to analyze complex problems and apply principles of computing and other relevant disciplines to identify solutions.
- Graduates will maintain and improve their skills as the computing field evolves, showing an ability to learn and adapt to new tools, technologies, and methodologies.
- Graduates will be effective team members and leaders, able to communicate and collaborate effectively in diverse teams to accomplish a common goal.
- Graduates will understand and address ethical, legal, security, and social issues and responsibilities relevant to software engineering.
- Graduates will be able to communicate effectively with a range of audiences about technical and non-technical issues.
- Graduates will demonstrate knowledge and understanding of project management principles and apply these to one's work, as either a member or leader of a team.
- Graduates will understand and apply quality assurance practices and methods in software development.
- Graduates will have the ability to design, develop, and integrate software systems within broader systems and networks.
- Lifelong Learning: Graduates will recognize the importance of lifelong learning and pursue ongoing professional development and education in their field.
- Global and Societal Impact: Graduates will understand the impact of software solutions in a global, economic, environmental, and societal context.
- Graduates will demonstrate creativity and innovation in solving software problems, and in designing software systems.

SYLLABUS:

UNIT	DETAILS	HOURS
I	**INTRODUCTION:** Introduction to Software Engineering, Definition of Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Applications, Software Myths. Software Development Life Cycle Model: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.	7

II	**SOFTWARE REQUIREMENT SPECIFICATIONS:** Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modeling, Data Flow Diagrams, Control Flow Model, SRS Document, IEEE Standards for SRS, Data Dictionary.	6
III	**SOFTWARE DESIGN:** Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Top-Down and Bottom-Up Design.	5
IV	**CODING & SOFTWARE TESTING & MAINTENANCE:** Top-Down and Bottom –Up programming, structured programming, Code Inspection, Compliance with Design and Coding Standards. Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Alpha and Beta Testing of Products. Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re-Engineering, Reverse Engineering.	8
V	**SOFTWARE MEASUREMENT & MATRICES:** Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs. Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management. Quality Assurance, Quality Control, Software Quality Attributes, Software Quality Assurance (SQA): Verification and Validation.	11
	TOTAL HOURS	37

Teacher Centric Approach **TC1: Chalk and Talk, TC2: PPT, TC3: Video Lectures TC4: Blended learning**
Learner Centric Approach: **LC1: Assignment. LC2: Mini project. LC3: Quiz/Class test. LC 4: Seminar on recent trends.** **LC5: Group Task. LC6: Others**

DETAILED SESSION PLAN

Lecture session/ Number	Topics to be covered	CO addressed	Teacher Centric Approach	Learner Centric Approach	References	Relevance with POs and PSOs
1	Introduction to Software	1	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	2
2	Software Characteristics, Software Crisis, Software	1	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3
3	Similarity and Differences from Conventional Engineering Processes, Applications,	1	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3
4	Software Myths. Software Development Life Cycle Model	1	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	2
5	Water Fall Model	1	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3
6	Prototype Model, Spiral Model	1	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3

7	Evolutionary Development Models, Iterative Enhancement Models.	1	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	2
8	Requirement Engineering Process	2	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3
9	Elicitation, Analysis, Documentation, Review	2	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3
10	Management of User Needs, Feasibility Study,	2	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	2
11	Information Modeling, Data Flow Diagrams, Control Flow	2	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3
12	SRS Document, IEEE Standards for SRS,	2	TC1, TC2	LC3	T1/T2/R1	2
13	Data Dictionary.	2	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3
14	Basic Concept of Software Design, Architectural Design	3	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3
15	Low Level Design: Modularization, Design Structure Charts	3	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	2
16	Flow Charts, Coupling and Cohesion Measures,	3	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	2
17	Design Strategies: Function Oriented Design	3	TC1, TC2	LC3	R1/R2/R3	2

18	Top-Down and Bottom-Up Design	3	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	3
19	Top-Down and Bottom –Up programming, structured programming	4	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	2
20	Code Inspection, Compliance with Design and Coding Standards.	4	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	3
21	Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing	4	TC1, TC2	LC1,LC2,LC3,LC4	R1/R2/R3	2
22	Regression Testing, Top-Down and Bottom-Up Testing Strategies	4	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	3
23	Test Drivers and Test Stubs,	4	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	2
24	Structural Testing (White Box Testing), Functional Testing (Black Box Testing),	4	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	3
25	Alpha and Beta Testing of Products Need for Maintenance,	4	TC1, TC2	LC3	R1/R2/R3	2

26	Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re-Engineering, Reverse Engineering.	4	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	2
27	Function Point (FP) Based Measures	5	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	3
28	Function Point (FP) Based Measures	5	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	3
29	Function Point (FP) Based Measures	5	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	2
30	Cyclomatic Complexity Measures: Control Flow Graphs.	5	TC1, TC2	LC1,LC2,LC3,LC4	R1/R2/R3	3
31	Cyclomatic Complexity	5	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	2
32	Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration,	5	TC1, TC2	LC1,LC2,LC3	R1/R2/R3	2
33	Estimation of Various Parameters	5	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	2
34	Constructive Cost Models (COCOMO), Resource Allocation Models,	5	TC1, TC2	LC3	T1/T2/R1	3
35	Constructive Cost Models	5	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3

36	Software Risk Analysis and Management. , Quality Assurance, Quality Control, Software Quality	5	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	2
37	Software Quality Assurance (SQA): Verification and Validation	5	TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3
38	Quiz		TC1, TC2	LC1,LC2,LC3	T1/T2/R1	3

TEXT/REFERENCE BOOKS:

T/R	
1	Pressman Roger S., "Software Engineering – A Practitioner's Approach", 6th Edition, McGraw Hill, 2004.
2	Sommerville Ian, Pearson Education, "Software Engineering", 5th edition, Addison Wesley, 1999.
3	Aggarwal KK, Singh, Yogesh, "Software Engineering", New Age International, 2000.
4	Jalote Pankaj,"An Integrated Approach to Software Engineering", 3rd edition, Narosa, 2005

 # WEB SOURCE REFERENCES (W):

1	https://www.javatpoint.com/software-engineering-tutorial
2	https://www.tutorialspoint.com/software_engineering/index.htm
3	W3schools.com

COURSE PRE-REQUISITES: A robust data infrastructure for efficient storage and management of diverse datasets. Advanced analytical tools and skilled personnel are essential for extracting meaningful insights.

C.CODE	COURSE NAME	DESCRIPTION	SEM
CS - 304	Software Engineering	This course provides a comprehensive overview of software development principles, methodologies, and best practices, emphasizing practical skills for designing and building software systems.	VI

COURSE OBJECTIVES:

1	To enable students to understand and apply software design principles, creating efficient, scalable, and maintainable software solutions.
2	To familiarize students with various software development methodologies like Agile, Waterfall, and DevOps, ensuring adaptability to industry practices.
3	To teach techniques for software testing, quality assurance, and debugging to produce reliable and error-free software.
4	To equip students with skills to gather, analyze, and manage software requirements effectively, ensuring alignment with stakeholder needs.

5	To introduce project management concepts, enabling students to lead software development projects, manage teams, and meet deadlines.

COURSE OUTCOMES:

S.NO	DESCRIPTION	PO(1..12) MAPPING	PSO(1..3) MAPPING
CO1	To learn the basic concepts of software engineering.	PO1, PO2, PO5, PO9	PSO1
CO2	To know about the requirements and process to engineer the software.	PO3, PO4, PO5, PO6	PSO3
CO3	To learn how to design a software & what are its strategies.	PO3, PO4, PO5, PO6	PSO3
CO4	To aware about the coding, testing & maintenance of software	PO10, PO11	PSO2, PSO3
CO5	To know about different metrics used for software evaluation.	PO1, PO2, PO5, PO9	PSO1
COURSE OVERALL PO/PSO MAPPING: 2/2			

COURSE OUTCOMES VS POs MAPPING (DETAILED; HIGH:3; MEDIUM:2; LOW:1**):**

S.NO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	1								2		2		
CO2	1	1	1								1		1	2	
CO3	2	1									1			1	1
CO4	1	1									1		1		1
CO5	1	1									1				1
CO1	1	1	1								2		2		

** For Entire Course, PO & PSO Mapping*

POs & PSO REFERENCE:

PO 1	Apply the knowledge of mathematics, science, engineering and Application fundamentals, and an engineering and Application specialization to the solution of complex engineering problems.	PO7	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	PSO1	To equip the students with theoretical and implementation knowledgebase in all the latest areas of Computer Science & Engineering for a successful career in software industries, pursuing higher studies, or entrepreneurial establishments.
PO 2	Identify, formulate, research literature, and analyze	PO8	Apply ethical principles and commit to professional ethics	PSO2	To nurture the students with the critical thinking abilities

	complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.		and responsibilities and norms of the engineering practice.		for better decision making by offering them a socially acceptable solutions to real life problems through computing paradigm.
PO 3	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.	PO9	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	PSO3	To nurture the students with the comprehensive analytical and design abilities by offering them techno-commercially feasible solutions of real business problems through computing.
PO 4	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.	PO10	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.		
PO 5	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.	PO11	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.		
PO 6	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.	PO12	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.		

COs VS POs MAPPING JUSTIFICATION:

S.NO	PO/PSO MAPPED	LEVEL OF MAPPING	JUSTIFICATION
CO1	2/1	1	It aligns with Program Outcome Level 1 by establishing a solid foundation in software development principles and essential concepts, preparing students for further specialized learning.
CO2	3/3	2	It goes beyond foundational concepts to explore advanced methodologies and practices, enhancing students' readiness for complex software development

			projects.
CO3	3/3	2	It delvs into advanced software engineering practices, methodologies, and project management, equipping students for challenging real-world software development scenarios.
CO4	2/3	1	It aligns with Program Outcome Level 1 by providing fundamental knowledge of software development principles, forming the basis for subsequent specialized learning in the program.
CO5	2/1	2	It offers advanced topics, methodologies, and practical experiences in software engineering, preparing students for complex projects and industry standards.

GAPS IN THE SYLLABUS - TO MEET INDUSTRY/PROFESSION REQUIREMENTS, POs & PSOs:

SNO	DESCRIPTION	PROPOSED ACTIONS
1	Lack of Agile and DevOps Integration: The syllabus may not adequately cover modern software development practices such as Agile methodologies and DevOps.	Introduce modules on Agile development, continuous integration, and deployment, aligning with industry demand for iterative and automated workflows.
2	Limited Focus on Software Security: The course may not emphasize cybersecurity and secure coding practices, leaving a gap in preparing students for industry security requirements.	Incorporate security-centric modules addressing secure coding, threat modeling, and vulnerability assessments to ensure software resilience.
3	Insufficient Exposure to Industry Tools: Students might lack familiarity with industry-standard development tools and version control systems.	Include hands-on training with tools like Git, JIRA, and industry-standard IDEs to enhance practical skills

		and toolset proficiency.
4	Limited Collaboration and Soft Skills: The syllabus may not emphasize collaboration, communication, and soft skills development, which are vital in real-world software engineering teams.	Introduce teamwork-oriented projects, communication workshops, and presentation skills training to prepare students for interdisciplinary collaboration.
5	Inadequate Exposure to Emerging Technologies: The course might not cover emerging technologies such as cloud computing, containerization, and microservices architecture.	Integrate modules on emerging technologies and trends to ensure students are familiar with industry-relevant advancements in software engineering.

PROPOSED ACTIONS: TOPICS BEYOND SYLLABUS/ASSIGNMENT/INDUSTRY VISIT/GUEST LECTURER/NPTEL ETC

 # TOPICS BEYOND SYLLABUS/ADVANCED TOPICS/DESIGN:

1	Containerization and Orchestration
2	Microservices Architecture
3	Serverless Computing
4	Continuous Integration/Continuous Deployment (CI/CD)
5	Ethical Hacking and Penetration Testing
6	AI and Machine Learning Integration
7	User Experience (UX) Design

DELIVERY/INSTRUCTIONAL METHODOLOGIES:

☐ CHALK & TALK	☐ STUD. ASSIGNMENT	☐ WEB RESOURCES	☐ NPTEL/OTHERS
☐ LCD/SMART BOARDS	☐ STUD. SEMINARS	☐ ADD-ON COURSES	☐ WEBNIARS

ASSESSMENT METHODOLOGIES-DIRECT

☐ ASSIGNMENTS	☐ STUD. SEMINARS	☐ TESTS/MODEL EXAMS	☐ UNIV. EXAMINATION
☐ STUD. LAB PRACTICES	☐ STUD. VIVA	☐ MINI/MAJOR PROJECTS	☐ CERTIFICATIONS
☐ ADD-ON COURSES	☐ OTHERS		

ASSESSMENT METHODOLOGIES-INDIRECT

☐ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE)	☐ STUDENT FEEDBACK ON FACULTY (TWICE)
☐ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS	☐ OTHERS

 # INNOVATIONS IN TEACHING/LEARNING/EVALUATION PROCESSES:

1. Implement a project-centric approach where students work on real-world software development projects throughout the course. This hands-on experience mirrors industry practices and allows for practical application of concepts.
2. Utilize a flipped classroom model, where students review theory online and attend in-person or virtual sessions for coding exercises, discussions, and problem-solving. This encourages active learning and peer collaboration.
3. Incorporate peer code review sessions, where students assess and provide feedback on each other's code. This promotes code quality, collaboration, and learning from diverse coding styles.
4. Organize hackathons or coding competitions within the course, challenging students to apply their software engineering skills under time constraints. This fosters creativity and teamwork.
5. Replace traditional exams with continuous assessment methods that involve solving real-world software engineering challenges. Evaluate students' problem-solving abilities, code quality, and project management skills.

Prepared by
Dr. Tapsi Nagpal

Approved by
A.Dean &HOD

 # Additionally, the details to be compiled separately by the Departmental Coordinator for the entire Department.