THOMAS ADEWUMI UNIVERSITY, OKO-IRESE	
Faculty	Computing and Applied Sciences
Department	Mathematical and Computing Science
Program	Computer Science
Course Code	CSC 403
Course Title	**SOFTWARE ENGINEERING**
Study Year	4
Credit Hours	3
Contact Hours	36
Pre-requisite	
Status	Compulsory
Semester	First
Mode of Assessment	Lecture, Assessment and Practical
Mode of Delivery	• Classroom Lectures • Laboratory Practical Sessions
Assignment **practical** **Test** **Examination** **Total**	10% 20% 70% 100%
Course Lecturer and Instructor	
Course Description	Software Engineering is a multidisciplinary course that focuses on the principles, practices, and techniques involved in designing, developing, testing, and maintaining software systems. It combines elements of computer science, mathematics, and engineering to address the challenges associated with building large-scale, complex software applications..
Course Objectives	This course would enable the understanding of the following: 1. Provide students with the required skills in software Engineering. 2. Teach students the software process and project management skills. 3. Teach students the specifications, requirements analysis and software design process. 4. Teach students how to test and implement software products.

	5. Teach students project management techniques.
Learning Outcome	At the end of the course, students will be able to: - Mention at least five(5) principles, and practices of software engineering. - List at least three(3) effective software requirements to meet user needs and project objectives. - Mention at least three(3) software design principles and techniques to create well-structured, modular, and maintainable software solutions. - Demonstrate proficiency in programming languages, development frameworks, and tools to implement software systems effectively. - Design and execute comprehensive software testing strategies to identify and rectify defects in software systems.
Detailed course contents	Software Design: Software architecture, Design Patterns, O. O. analysis & Design, Design for re-use. Using APIS: API programming Class browsers and related tools, Component-based computing. Software tools and Environment: Requirements analysis and design modelling Tools, Testing tools, Tool integration mech. Team Management, Project Scheduling, Software measurement and estimation techniques, Risk analysis, Software quality assurance, Software Configuration Management, Project Management tools.

Course Contents Sequencing

Weeks	Detailed Course Outline	Allocated Time
WEEK 1	Introduction to Software Engineering: • Definition and scope of software engineering • Software engineering principles and practices • Importance of software engineering in modern society	3 Hours
WEEK 2	Software Development Life Cycle (SDLC): • Overview of different SDLC models (e.g., Waterfall, Agile, Spiral) • Phases of the SDLC: requirements analysis, design, implementation, testing, deployment, maintenance • Comparison of different SDLC models and their suitability for various projects	3 Hours
WEEK 3, 4	Software Requirements Engineering:	6 Hours

	• Importance of requirements engineering • Techniques for gathering requirements (interviews, surveys, observations) • Requirements documentation and specification • Requirements validation and management **C.A Test**	
WEEK 5, 6	Software Design Principles: • Design principles and concepts (e.g., modularity, cohesion, coupling) • Architectural patterns and design patterns (e.g., MVC, Observer, Factory) • UML (Unified Modeling Language) diagrams for software design	6 Hours
WEEK 7,8	Software Testing and Quality Assurance: • Importance of software testing and quality assurance • Types of testing (unit testing, integration testing, system testing, acceptance testing) • Test-driven development (TDD) and automated testing • Software quality metrics and measurement techniques	6 Hours
WEEK 9, 10	Software Project Management: • Project planning and estimation techniques • Project scheduling and resource allocation • Risk management and mitigation strategies • Team collaboration and communication tools **C.A Test**	6 Hours
WEEK 11	Software Maintenance and Evolution: • Challenges and importance of software maintenance • Bug tracking and debugging techniques • Software updates and version control • Software documentation and knowledge management	3 Hours
WEEK 12	Software Ethics and Professionalism: • Ethical considerations in software engineering • Intellectual property and copyright issues • Professional responsibilities and codes of conduct	3 Hours

	• Social implications and privacy concerns in software development	
WEEK 13	REVISION	

READING LIST:

- Agile Software Development: Principles, Patterns, and Practices by Robert C. Martin
- Software Engineering: A Methodical Approach by Elvis C. Foster and Dennis A. Frailey
- Requirements Engineering: From System Goals to UML Models to Software Specifications by Axel van Lamsweerde
- Software Engineering: A Concise Study by Pankaj Jalote
- Software Engineering: Modern Approaches by Eric J. Braude and Michael E. Bernstein
- Software Engineering: Theory and Practice by Shari Lawrence Pfleeger and Joanne M. Atlee