Kalinga University
Technology
Master Of Technology In Computer Science
(Object Oriented Software Development)

PO

S. No.	Program Outcome (PO) Description
1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PSO

S. No.	Program Specific Outcome (PSO) Description
1	Model computational problems by applying mathematical concepts and design solutions using suitable data structures and algorithmic techniques
2	Design and develop solutions by following standard software engineering principles and implement by using suitable programming languages and platforms
3	Develop system solutions involving both hardware and software modules
4	Understand, Analyze and Develop computer programs for efficient design of computer-based systems of varying complexity.
5	Demonstrate basic knowledge of computer applications and apply standard practices in software project development.

CO

S.No.	Course Code	Course Name	Course Outcome (CO's) - Description
1	MTOOSD101	Advanced Engineering Mathematics	**CO1:** To know how root finding techniques can be used to solve practical engineering problems.
			CO2: To apply the analytical technique to express periodic function as a Fourier sine and cosine series.
			CO3: To apply partial differential techniques to solve the physical engineering problems.
			CO4: To apply the concept of probability to find the physical significance of various distribution phenomena
2	MTOOSD102	Object Oriented Analysis And Design Using UML	**CO1:** An ability to analyze the problem and apply to real world problems.
			CO2: An ability to apply knowledge of OOPs concepts in Object Oriented Design.
			CO3: An ability to analyze the case study and apply the UML notations
			CO4: An ability to gather functional requirements and apply the use case diagrams
			CO5: Draw activity and state chart diagram for real word applications
			CO6: Apply a class diagram and object diagram for user requirements
			CO7: Apply component and deployment diagram for based on Requirements
			CO8: take the case studies and model it in different views with respect user requirement such as use case, logical, component and deployment and etc, and preparation of document of the project for the unified Library application.
3	MTOOSD103	Advanced Operating System	**CO1:** Analyze the structure of OS and basic architectural components involved in OS design
			CO2: Analyze and design the applications to run in parallel either using process or thread models of different OS
			CO3: Analyze the various device and resource management techniques for timesharing and distributed systems
			CO4: Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
			CO5: Interpret the mechanisms adopted for file sharing in distributed Applications
			CO6: Conceptualize the components involved in designing a contemporary OS
4	MTOOSD104	Internet and Web Technology	**CO1:** Understand the enabling technologies for building Internet and Web database applications.
			CO2: Understand the different components for developing client/server applications.
			CO3: Apply the techniques and features of the client/server development languages to construct a database application based on Internet.
			CO4: Develop the web database applications through programming exercises. Implement Web Services through technologies.
			CO1: Able to understand the use of OOPs concepts.
			CO2: Able to solve real world problems using OOP techniques
			CO3: Able to understand the use of abstraction

5	MTOOSD105	Java Programming & Application	**CO4:**	Able to understand the use of Packages and Interface in java.
			CO5:	Able to develop and understand exception handling, multithreaded applications with synchronization.
			CO6:	Able to understand the use of Collection Framework
			CO7:	Able to design GUI based applications and develop applets for web applications
6	MTOOSD201	Compiler Design	**CO1:**	To realize basics of compiler design and apply for real time applications.
			CO2:	To introduce different translation languages
			CO3:	To understand the importance of code optimization
			CO4:	To know about compiler generation tools and techniques
			CO5:	To learn working of compiler and non compiler applications
			CO6:	Design a compiler for a simple programming language
7	MTOOSD202	Object Oriented Software Engineering	**CO1:**	Extract an Object Model and Dynamic Model of system functionality and performance from the requirements
			CO2:	Design and implement structured, robust, maintainable object-oriented systems across multiple platforms and appropriate programming languages from the specifications developed.
			CO3:	Develop teamwork and management skills to divide tasks and effectively develop projects in large software teams.
			CO4:	Produce industry standard documentation from requirements, analysis, and design through testing and verification.
8	MTOOSD203	Analysis & Design of Algorithms	**CO1:**	Argue the correctness of algorithms using inductive proofs and invariants.
			CO2:	Analyze worst-case running times of algorithms using asymptotic analysis.
			CO3:	Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm.
			CO4:	Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms
			CO5:	Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm.
			CO6:	Synthesize dynamic-programming algorithms, and analyze them.
			CO7:	Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm.
			CO8:	Synthesize greedy algorithms, and analyze them.
9	MTOOSD204	Distributed Operating System	**CO1:**	Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
			CO2:	Learn the various resource management techniques for distributed systems
			CO3:	Identify the different features of real time and mobile operating systems
			CO4:	Modify existing open source kernels in terms of functionality or features used.

10	MTOOSD205	Design & Development of Cloud Applications	**CO1:**	Understanding the key dimensions of the challenge of Cloud Computing
			CO2:	Assessment of the economics , financial, and technological implications for selecting cloud Computing for own organization
			CO3:	Assessing the financial, technological, and organizational capacity of employer's for actively initiating and installing cloud-based applications
			CO4:	Assessment of own organizations' needs for capacity building and training in cloud computing-related IT areas
11	MTOOSD301	Information Storage & Management	**CO1:**	Search, retrieve and synthesize information from a variety of systems and sources.
			CO2:	Evaluate systems and technologies in terms of quality, functionality, cost-effectiveness and adherence to professional standards
			CO3:	Integrate emerging technologies into professional practice
			CO4:	Apply theory and principles to diverse information contexts
12	MTOOSD302	Communication & Research Methodology	**CO1:**	Overall communication improvement which results in high quality communication.
			CO2:	Improved report writing skill to make your reports more attractive.
			CO3:	Get awareness on the various aspects of research.
			CO4:	Improved data and result representation.
13	MTOOSD303A	Advanced Computer Architecture	**CO1:**	Understand the Concept of Parallel Processing and its applications
			CO2:	Implement the Hardware for Arithmetic Operations
			CO3:	Analyze the performance of different scalar Computers
			CO4:	Develop the Pipelining Concept for a given set of Instructions
			CO5:	Distinguish the performance of pipelining and non pipelining environment in a processor
14	MTOOSD303B	Advanced Software Project Management	**CO1:**	Learn and become familiar with advanced terminology, concepts, and definitions
			CO2:	Gain an understanding of key project success and failure factors within a project.
			CO3:	Develop the mindset, key skills and processes to manage project teams
			CO4:	Be able to refine the project scope statement
			CO5:	Determine the approach for decomposing the work of the project to the required level of detail
			CO6:	Be able to apply key project managements