



Shri. Gajanan Maharaj Shikshan Prasarak Mandal's
Sharadchandra Pawar College of Engineering
Dumbarwadi(Otur), Tal- Junnar, Dist-Pune 410504

POs:

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, 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.

| | | |
|--------|---------------------------------|--|
| 210241 | Discrete Mathematics | <p>Course Outcomes:</p> <ol style="list-style-type: none"> 1. To introduce students to understand, explain, and apply the foundational mathematical concepts at the core of computer science. 3. To understand use of set, function and relation models to understand practical examples ,and interpret the associated operations and terminologies in context. 4. To acquire knowledge of logic and proof techniques to expand mathematical maturity. 5. To learn the fundamental counting principle, permutations, and combinations. 6. To study how to model problem using graph and tree. 7. To learn how abstract algebra is used in coding theory. <p>Course Outcomes:</p> <ol style="list-style-type: none"> 1. Formulate problems precisely, solve the problems, apply formal proof techniques, and 2. explain the reasoning clearly. 3. Apply appropriate mathematical concepts and skills to solve problems in both familiar and unfamiliar situations including those in real-life contexts. 4. Design and analyze real world engineering problems by applying set theory, propositional logic and to construct proofs using mathematical induction. 5. Specify, manipulate and apply equivalence relations; construct and use functions and apply these concepts to solve new problems. 6. Calculate numbers of possible outcomes using permutations and combinations; to model and analyze computational processes using combinatorics. 7. Model and solve computing problem using tree and graph and solve problems using 8. appropriate algorithms. 9. Analyze the properties of binary operations, apply abstract algebra in coding theory and evaluate the algebraic structures. |
| 210242 | Fundamentals of Data Structures | <p>Course Objective:</p> <ol style="list-style-type: none"> 1. To understand the standard and abstract data representation methods. 2. To acquaint with the structural constraints and advantages in usage of the data. 3. To understand various data structures, operations on it and the memory requirements 4. To understand various data searching and sorting methods. 5. To understand various algorithmic strategies to approach the problem solution. <p>Course Outcomes:</p> <ol style="list-style-type: none"> 1. Design the algorithms to solve the programming problems, identify appropriate algorithmic strategy for specific application, and analyze the time and space complexity. |

| | | |
|--------|-----------------------------------|--|
| | | <ol style="list-style-type: none"> Discriminate the usage of various structures, Design/Program/Implement the appropriate data structures; use them in implementations of abstract data types and Identity the appropriate data structure in approaching the problem solution. Demonstrate use of sequential data structures- Array and Linked lists to store and process data. Understand the computational efficiency of the principal algorithms for searching and sorting and choose the most efficient one for the application. Compare and contrast different implementations of data structures (dynamic and static). Understand, Implement and apply principles of data structures-stack and queue to solve computational problems. |
| 210243 | Object Oriented Programming (OOP) | <p>Course Objectives:</p> <ol style="list-style-type: none"> To learn the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. To learn the syntax and semantics of the C++ programming language. To understand the concept of data abstraction and encapsulation, how to design C++ classes for code reuse, how to implement copy constructors and class member functions, to overload functions and operators in C++. To learn how inheritance and virtual functions implement dynamic binding with polymorphism. To learn how to design and implement generic classes with C++ templates and how to use exception handling in C++ programs. <p>Course Outcomes:</p> <ol style="list-style-type: none"> Apply constructs- sequence, selection and iteration; classes and objects, inheritance, use of predefined classes from libraries while developing software. Design object-oriented solutions for small systems involving multiple objects. Use virtual and pure virtual function and complex programming situations. Apply object-oriented software principles in problem solving. Analyze the strengths of object-oriented programming. Develop the application using object oriented programming language(C++). |

| | | |
|--------|--------------------------------------|---|
| 210244 | Computer Graphics | <p>Course-Objectives :</p> <ol style="list-style-type: none"> 1. Remembering: To acquaint the learner with the basic concepts of Computer Graphics. 2. Understanding: To learn the various algorithms for generating and rendering graphical figures. 3. Applying: To get familiar with mathematics behind the graphical transformations. 4. Understanding: To understand and apply various methods and techniques regarding projections, animation, shading, illumination and lighting. 5. Creating: To generate Interactive graphics using OpenGL. <p>Course Outcomes:</p> <ol style="list-style-type: none"> 1. Identify the basic terminologies of Computer Graphics and interpret the mathematical 2. foundation of the concepts of computer graphics. 3. Apply mathematics to develop Computer programs for elementary graphic operations. 4. Illustrate the concepts of windowing and clipping and apply various algorithms to fill and clip polygons. 5. Understand and apply the core concepts of computer graphics, including transformation in two and three dimensions, viewing and projection. 6. Understand the concepts of color models, lighting, shading models and hidden surface elimination. 7. Create effective programs using concepts of curves, fractals, animation and gaming. |
| 210245 | Digital Electronics and Logic Design | <p>Course Objectives:</p> <ol style="list-style-type: none"> 1. To study number systems and develop skills for design and implementation of combinational logic circuits and sequential circuits 2. To understand the functionalities, properties and applicability of Logic Families. 3. To introduce programmable logic devices and ASM chart and synchronous state machines. 4. To introduce students to basics of microprocessor. <p>Course Outcomes:</p> <ol style="list-style-type: none"> 1. Simplify Boolean Expressions using K Map. 2. Design and implement combinational circuits. 3. Design and implement sequential circuits. 4. Develop simple real-world application using ASM and PLD. 5. Differentiate and Choose appropriate logic families IC packages as per the given design specifications. 6. Explain organization and architecture of computer system |

| Course Code | Course Name | Course Outcomes |
|-------------|--------------------------------|---|
| 207003 | Engineering Mathematics III | <p>Course Objectives:</p> <ol style="list-style-type: none"> 1. Linear differential equations of higher order applicable to Control systems, Computer vision and Robotics. 2. Transform techniques such as Fourier transform, Z-transform and applications to Image processing. 3. Statistical methods such as correlation, regression analysis and probability theory to analyze data and to make predictions applicable to machine intelligence. 4. Vector calculus necessary to analyze and design complex electrical and electronic devices as appropriate to Computer engineering. <p>Program- Outcomes:</p> <ol style="list-style-type: none"> 1. Solve Linear differential equations, essential in modelling and design of computer-based systems. 2. Apply concept of Fourier transform and Z-transform and its applications to continuous and discrete systems and image processing. 3. Apply Statistical methods like correlation and regression analysis and probability theory for data analysis and predictions in machine learning. 4. Solve Algebraic and Transcendental equations and System of linear equations using numerical techniques. 5. Obtain Interpolating polynomials, numerical differentiation and integration, numerical solutions of ordinary differential equations used in modern scientific computing. |
| 210252 | Data Structures and Algorithms | <p>Course-Objectives :</p> <ol style="list-style-type: none"> 1. To develop a logic for graphical modeling of the real life problems. 2. To suggest appropriate data structure and algorithm for graphical solutions of the problems. 3. To understand advanced data structures to solve complex problems in various domains. 4. To operate on the various structured data 5. To build the logic to use appropriate data structure in logical and computational solutions. 6. To understand various algorithmic strategies to approach the problem solution. <p>Program-Outcomes :</p> <ol style="list-style-type: none"> 1. Identify and articulate the complexity goals and benefits of a good hashing scheme for real-world applications. 2. Apply non-linear data structures for solving problems of various domain. 3. Design and specify the operations of a nonlinear-based abstract data type and implement them in a high-level programming language. 4. Analyze the algorithmic solutions for resource requirements and optimization 5. Use efficient indexing methods and multiway search techniques to store and maintain data. 6. Use appropriate modern tools to understand and analyze the functionalities confined to the secondary storage. |

| | | |
|--------|----------------------|---|
| 210253 | Software Engineering | <p>Course Objectives:</p> <ol style="list-style-type: none"> 1. To learn and understand the principles of Software Engineering. 2. To be acquainted with methods of capturing, specifying, visualizing and analyzing software requirements. 3. To apply design and testing principles to software project development. 4. To understand project management through life cycle of the project. <p>Course Outcomes:</p> <ol style="list-style-type: none"> 1. Analyze software requirements and formulate design solution for a software. 2. Design applicable solutions in one or more application domains using software engineering approaches that integrate ethical, social, legal and economic concerns. 3. Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society in all aspects and evolving into their continuous professional development. 4. Model and design User interface and component-level. 5. Identify and handle risk management and software configuration management. 6. Utilize knowledge of software testing approaches, approaches to verification and validation. 7. Construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain efficient, reliable, robust and cost-effective software solutions. |
| 210254 | Microprocess or | <p>Course-Objectives :</p> <ol style="list-style-type: none"> 1. To learn and distinguish the architecture and programmer's model of advanced processor. 2. To identify the system level features and processes of advanced processors. 3. To acquaint the learner with application instruction set and logic to build assembly language programs. <p>Program-Outcomes :</p> <ol style="list-style-type: none"> 1. Exhibit skill of assembly language programming for the application. 2. Classify Processor architectures. 3. Illustrate advanced features of 80386 Microprocessor. 4. Compare and contrast different processor modes. 5. Use interrupts mechanism in applications 6. Differentiate between Microprocessors and Microcontrollers. 7. Identify and analyze the tools and techniques used to design, implement, and debug microprocessor-based systems. |

| | | |
|--------|-------------------------------------|--|
| 210255 | Principles of Programming Languages | <p>Course Objectives:</p> <ol style="list-style-type: none"> 1. To learn basic principles of programming languages and programming paradigms. 2. To learn structuring the data and manipulation of data, computation and program structure. 3. To learn Object Oriented Programming (OOP) principles using Java Programming Language. 4. To learn basic concepts of logical and functional programming language. <p>Course Outcomes:</p> <ol style="list-style-type: none"> 1. Make use of basic principles of programming languages. 2. Develop a program with Data representation and Computations. 3. Develop programs using Object Oriented Programming language : Java. 4. Develop application using inheritance, encapsulation, and polymorphism. 5. Demonstrate Multithreading for robust application development. 6. Develop a simple program using basic concepts of Functional and Logical programming paradigm. |
|--------|-------------------------------------|--|