FACULTY OF ENGINEERING & TECHNOLOGY

BACHELOR OF ENGINEERING

MINOR DEGREE IN COMPUTER PROGRAMMING (DRAFT SYLLABUS)

Course Structure

Sr. No.	Semester	Temp. Course Code	Course Title	L	T	P	Credits
1.	**3**	**CP-1**	**Advanced C Programming**	**3**	**0**	**2**	**4**
2.	**4**	**CP-2**	**Database Management System**	**3**	**0**	**2**	**4**
3.	**5**	**CP-3**	**Python Programming**	**3**	**0**	**2**	**4**
4.	**6**	**CP-4**	**Web Technologies**	**3**	**0**	**2**	**4**
5.	**6**	**CP-5**	**Mini Project**	**0**	**0**	**4**	**2**
TOTAL				**12**	**0**	**12**	**18**

Detailed Syllabus

Course code:	**CP-1**
Name of the course:	Advanced C Programming
Semester:	**3**
Category of Course:	CP

Course Objectives:

This Course intends to develop programming skills in the students, using a popular structured programming language ‘C’. The students will learn step by step procedure (i.e. flowcharting &Algorithm)of any program development process. The programming skills thus acquired using ‘C' language can be used for acquiring necessary programming skill to work with advance level programming languages which in turn will be helping in developing programs for the scientific, research and business purposes.

Teaching & Examination Scheme:

Teaching Scheme			Credits	Examination Marks (Maximum/Passing)		Total
				External		
L	T	P	C	ESE(T)	ESE(P)	
3	0	2	4	50/25	50/25	100 /50

ESE(T): End Semester Examination(Theory)

ESE(P): End Semester Examination(Practical)

Course Contents:

Unit	Contents	Hours
1	Overview of C and Managing input & Output operations	5
2	Decision making, branching, Arrays &Strings.	6
3	User defined functions, Structure, Unions and Pointers	6
4	File Management , Dynamic Memory Allocation & Linked Lists	7
5	Stacks, Queues Trees& Graphs	6

Suggested Specification table with Marks (Theory)(Revised Bloom's Taxonomy):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
15%	25%	20%	15%	15%	10%

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

1.Programming in ANSI C by Balaguruswamy

2.C Programming: Test Your Skills, 1/ebyAshokKamthane

3.Programmingin C Ansi standard, by YashwantKanetkar

4. Programmingwith C, Gottfried, McGraw-Hill.

5. An Introduction to Data Structures with Applications. by Jean-Paul Tremblay & Paul G. Sorenson Publisher-Tata McGrawHill.

6. Data Structures using C & C++ -By Ten Baum Publisher –Prenctice-Hall International.

7. Data Structures: A Pseudo-code approach with C -By Gilberg&Forouzan Publisher-Thomson Learning.

Course Outcomes (CO):

Sr. No.	Course Outcome Statements	% weightage
1	Derive solutions for the given problems using decision making and branching concepts.	10
2	Implement the concept of arrays& strings.	10
3.	Implement problem solving skills using pointer & File management concept of the programming languages.	15
4.	Learn the functions of Structures, Unions and Pointers.	15
5.	Efficiently use data structures for problem solving and Understandthe features of Stacks and Queues.	25
6.	Know the different kinds of trees and their representations and Understand the creation, traversal and representation of graphs.	25

List of Practicals /Tutorials:

1. Write C programs to implement basic arithmetic operations –sum, average, product, difference, quotient and remainder of given numbers etc.

2. Write a C program to find largest and smallest of given numbers

3. Write a C program to find Addition of Two Matrices

4. Write a C program to find Multiplication of Two Matrices

5. Write a C program to print 2-D array using pointers.

6. Write a C program to allocate memory dynamically using memory allocation functions (malloc, calloc, realloc, free).

7. Write a C program to copy one file to another.

8. Write a C program to display the contents of a file.

9. Write a C program to merge two files into a third file.

10. Write a program that uses functions to perform the following operations on singly linked list i) Creation ii) Insertion iii) Deletion iv) Traversal.

11. Write a program that implements the following sorting i) Bubble sort ii) Selection sort.

12. Write a program which create binary search tree.

Supplementary Learning Material:

1. https://onlinecourses.nptel.ac.in/noc21_cs01/preview

2. https://www.programiz.com/c-programming

3. https://www.cprogramming.com/advtutorial.html

Course Articulation Matrix:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
1	3	3	3	3	1	3	3	1	3	1	3	1	3	2	2
2	3	1	-	-	1	1	-	-	1	-	-	1	3	1	1
3	3	1	1	3	1	1	1	-	1	-	1	1	3	1	1
4	3	1	-	1	1	1	-	-	1	-	-	1	3	1	1
5	3	1	1	-	3	3	1	-	3	1	3	1	3	2	2
6	3	3	1	-	1	1	-	-	1	3	-	1	3	2	2

1. Slight (Low) 2. Moderate (Medium) 3. Substantial (High) '–' No correlation

****

Detailed Syllabus

Course code:	**CP-2**
Name of the course:	Database Management System
Semester:	**4**
Category of Course:	**CP**

Course objectives:

The course, Database Management Systems, provides an introduction to the management of database systems. The course emphasizes the understanding of the fundamentals of relational systems including data models, database architectures, and database manipulations. The course uses a problem-based approach to learning.

Teaching & Examination Scheme:

Teaching Scheme			Credits	Examination Marks (Maximum/Passing)		Total
L	T	P	C	External		
				ESE(T)	ESE(P)	
3	0	2	4	50/25	50/25	100 /50

ESE(T): End Semester Examination(Theory)

ESE(P): End Semester Examination(Practical)

Course Contents:

Unit	Contents	Hours
1	Course Overview. Introduction to RDBMS & Structured Query Language (SQL)	8
2	Relational Algebra, Entity-Relationship Model& Relational Database Design	5
3	Application Development. Case Studies. Storage, File Structure, Indexing and Hashing &Query Processing	5
4	Query Optimization,Transactions (Serializability and Recoverability),	6
5	Concurrency Control, Recovery Systems and Course Summarization.	6

Suggested Specification table with Marks (Theory)(Revised Bloom's Taxonomy):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
15%	25%	20%	15%	15%	10%

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create

Reference Books:

-Database System Concepts by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, 6th Edition, McGraw-Hill Education, 2010.

Course Outcomes (CO):

Sr. No.	**Course Outcome Statements**	**% weightage**
1	State the importance of DBMS and describe fundamental elements of a relational data models.	10
2	Master the basics of SQL and construct queries using SQL, Relational Algebra and Calculus and apply query processing.	15
3.	Improve the schemas using normal forms address the problem s like decomposition, functional dependency and redundancy.	15
4.	Identify the basic issues of transaction processing and recovery techniques.	15
5.	Identify the basic issues of concurrency control and distributed database.	20
6.	Enhance team work by design and development of a database application system.	25

List of Practicals /Tutorials:

1. To study DDL-create and DML-insert commands

2. Create table and insert sample data in tables.

3. Perform queries involving predicates LIKE, BETWEEN, IN etc.

4. To perform various data manipulation commands, aggregate functions and sorting concept on all created tables.

5. To study Single-row functions.

6. Displaying data from Multiple Tables (join)

7. To apply the concept of Aggregating Data using Group functions.

8. To solve queries using the concept of sub query.

9. To apply the concept of security and privileges

10. To study Transaction control commands

11. Write Cursor

12. Write Trigger

Supplementary Learning Material:

1. https://www.tutorialspoint.com/dbms/
2. https://www.w3schools.com/sql/
3. https://www.codecademy.com/learn/learn-sql
4. https://in.udacity.com/

Course Articulation Matrix:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
1	-	-	-	-	-	-	-	-	-	-	-	-	3	3	2
2	-	2	-	1	2	1	-	-	1	-	3	3	3	3	2
3	-	2	-	1	2	-	-	1	1	1	3	3	3	3	2
4	-	2	-	-	-	1	-	1	1	1	-	-	3	3	2
5	-	2	-	-	-	-	-	-	-	-	-	-	3	3	2
6	-	-	2	2	-	1	-	-	2	-	3	-	3	3	2

1. Slight (Low) 2. Moderate (Medium) 3. Substantial (High) ‘–‘ No correlation

****

Detailed Syllabus

Course code:	**CP-3**
Name of the course:	**Python Programming**
Semester:	**5**
Category of Course:	**CP**

Course objectives:

This course is an introduction to programming and problem solving in Python. It does not assume any prior knowledge of programming. Using some motivating examples, the course quickly builds up basic concepts such as conditionals, loops, functions, lists, strings and tuples. It goes on to cover searching and sorting algorithms, dynamic programming and backtracking, as well as topics such as exception handling and using files. As far as data structures are concerned, the course covers Python dictionaries as well as classes and objects for defining user defined datatypes such as linked lists and binary search trees.

Teaching & Examination Scheme:

Teaching Scheme			Credits	Examination Marks(Maximum/Passing)		Total
L	T	P	C	External		
				ESE(T)	ESE(P)	
3	0	2	4	50/25	50/25	100 /50

ESE(T): End Semester Examination (Theory)

ESE(P): End Semester Examination (Practical)

Course Contents:

Sr. No.	**Content**	**Total Hrs**
1	Informal introduction to programming, algorithms and data structures, Installing Python, gcd in Python: variables, operations, control flow - assignments, condition-als, loops, functions Python: types, expressions, strings, lists, tuples, Python memory model: names, mutable and immutable valuesList operations: slices etc. Binary search, Inductive function definitions: numerical and structural induction, Elementary inductive sorting: selection and insertion sort, In-place sorting	07
2	Basic Algorithmic analysis, asymptotic complexity, O() notation, Arrays vs lists, Merge sort, Quicksort, Stable sorting, Dictionaries, More on Python functions: optional arguments, default values, Passing functions as arguments, Higher order functions on lists: map, filter, list comprehension	06

3	Exception handling, Basic input/output, Handling files, String processing, Backtracking: N Queens, recording all solutions, Scope in Python: local, global, nonlocal names, Nested functions, Data structures: stack, queue, Heaps	06
4	Abstract data types, Classes and objects in Python, "Linked" lists: find, insert, delete, Binary search trees, Height-balanced binary search trees	06
5	Efficient evaluation of recursive definitions: memorization, Dynamic programming, C and manual memory management, Other programming paradigms: functional programming	05

Suggested Specification table with Marks (Theory) (Revised Bloom's Taxonomy):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
10%	20%	20%	15%	15%	20%

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

1. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India
2. R. NageswaraRao, "Core Python Programming", dreamtech
3. Wesley J. Chun. "Core Python Programming - Second Edition", Prentice Hall
4. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms in Python", Wiley
5. Kenneth A. Lambert, "Fundamentals of Python – First Programs", CENGAGE Publication
6. Luke Sneeringer, "Professional Python", Wrox

Course Outcomes (CO):

Sr. No.	**Course Outcome Statements**	**% weightage**
1	To develop proficiency in creating based applications using the Python Programming Language	5
2	To be able to understand the various data structures available in Python programming language and apply them in solving computational problems.	35
3	To be able to do testing and debugging of code written in Python.	15

4	To be able to do text filtering with regular expressions in Python	25
5	To be able to create socket applications in Python	10
6	To be able to perform various analyzing between programming codes	10

List of Practicals /Tutorials:

1. Develop programs to understand the control structures of python
2. Develop programs to learn different types of structures (list, dictionary, tuples) in python
3. Write function to compute gcd, lcm of two numbers.
4. Develop programs to learn concept of functions scoping, recursion and list mutability.
5. Develop programs to understand working of exception handling and assertions.
6. Write a program to compute the number of characters, words and lines in a file**.**
7. Develop programs for data structure algorithms using python – searching, sorting and hash tables.
8. Develop programs to learn regular expressions using python.
9. Write a program to print each line of a file in reverse order. Write a program to compute the number of characters, words and lines in a file.
10. Python program for implementation of Insertion Sort
11. Python program for implementation of Quick Sort
12. Write a script that imports requests and fetch content from the page

Supplementary Learning Material:

1.https://docs.python.org/
2. https://onlinecourses.nptel.ac.in/noc20_cs70/preview
3. https://realpython.com/tutorials/advanced/
4. https://www.python-course.eu/advanced_python.php

Course Articulation Matrix:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO 10**	**PO 11**	**PO 12**	**PSO 13**	**PSO 14**	**PSO 15**
1	-	-	-	1	1	-	-	-	-	-	-	-	3	2	2
2	-	3	3	3	3	3	1	-	-	-	1	1	2	1	2
3	1	2	2	1	-	-	-	1	2	-	-	-	3	2	2
4	-	3	3	2	3	-	-	-	-	-	-	-	3	2	2
5	-	-	2	-	-	-	-	-	-	-	-	1	3	2	3
6	1	-	-	-	1	-	-	-	1	-	-	-	3	2	2

1. Slight (Low) 2. Moderate (Medium) 3. Substantial (High) '–' No correlation

****

Detailed Syllabus

Course code:	**CP-4**
Name of the course:	**Web Technologies**
Semester:	**6**
Category of Course:	**CP**

Course objectives:

In the era of digitization, the demand of Internet based applications is increasing day by day. To put students in the orbit of this Internet driven world and to make them comfortable in developing various web-based applications, this course is focusing on front-end and Back-end design.

Teaching & Examination Scheme:

Teaching Scheme			Credits	Examination Marks(Maximum/Passing)		Total
				External		
L	T	P	C	ESE(T)	ESE(P)	
3	0	2	4	50/25	50/25	100 /50

ESE(T): End Semester Examination (Theory)

ESE(P): End Semester Examination (Practical)

Course Contents:

Unit	Contents	Hours
1	**Introduction:** Basics of WWW, HTTP protocol methods and headers, HTTP Request and Response, Architecture of web browser, Web server installation and configuration.	03
2	**HTML:** HTML page structure, formatting tags in HTML, tables, links, images, meta tags, frames, html form tags, media, APIs, HTML5 tags in relation to validations and SEO. **CSS:** Need for CSS, Basic syntax and structure, Backgrounds, Colors and properties, manipulating texts, Fonts, borders and boxes, Margins, Padding Lists, CSS2, CSS3, Animations, Tool-Tips, Style images, Variables, Flex Box, Media Queries, Wildcard Selectors (*, ^ and $) in CSS, Working with Gradients, Pseudo Class, Pseudo elements, basic of frameworks like Bootstrap, Responsive web design and Media Query, CSS variables	07
3	**Java Script:** Javascript Syntax, Types of Javascript, variables, arrays, functions,	07

	conditions, loops, Pop up boxes, Javascript objects and DOM, Javascript inbuilt functions, Javascript validations, Regular expressions, Event handling with Javascript, Callbacks in Javascript, Function as arguments in Javascript, Object concepts in Javascript, JSON **PHP Basics** Introduction to Server-side programming, PHP variables, decision and looping with examples, PHP and HTML, Arrays, Functions, Browser control and detection, String, Form processing, File uploads, Dates and timezone, Working with Regular Expressions, Exception Handling, Working with JSON data, Object Oriented Programming with PHP	
4	**Session and State Management using PHP** Need of session management, Various techniques for state and session management like: Hidden Fields, Query String, Cookie and Session **Database Connectivity using PHP**: Basic commands for database connection and query execution with CURD examples, Object oriented database access using PHP	07
5	**Advanced Concepts:** Asynchronous Web requests using AJAX, Creating REST API using PHP **JQuery:** Working with jQuery, using plugins in jQuery and Creating Image slider, generating charts from data using 3rd Party Libs	06

Suggested Specification table with Marks (Theory)(Revised Bloom's Taxonomy):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
10%	20%	20%	16%	14%	20%

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

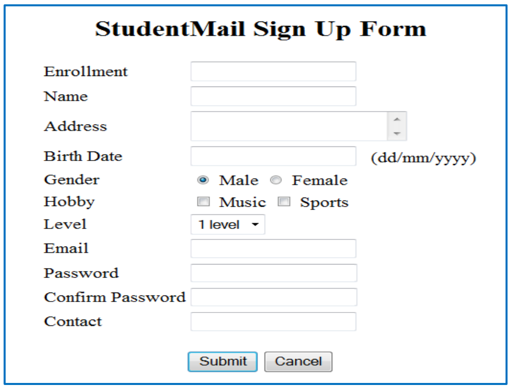
1. HTML 5 Black Book 2Ed, by Kogent Learning Solutions Inc.
2. Learning PHP, MySQL, JavaScript, CSS & HTML5, 3rd Edition
3. A Step-by-Step Guide to Creating Dynamic Websites By Robin Nixon Publisher: O'Reilly Media
3. JavaScript for impatient programmers by Dr. Axel Rauschmayer
4. PHP: The Complete Reference By Steven Holzner, McGrawhill

Course Outcomes (CO):

Sr. No.	Course Outcome Statements	% weightage
1	Understand the concepts of WWW, HTTP protocol and client-server architecture.	5
2	Learn and apply various HTML tags to build the user-friendly web pages.	10
3	Explore the new features of CSS to define and apply CSS rules in the web pages for rich User Interface.	15
4	Create interactive web pages to improve the user experience using client-side scripting with Javascript.	25
5	Design and develop fully functional dynamic web applications using the concepts of PHP, MySQL,	35
6	Learn and apply advanced asynchronous web communication mechanisms like REST API, AJAX and JQuery for building highly interactive webpages	10

List of Practicals/Tutorials:

1. Introduction to HTML5 different tags with syntax, example and description
2. Design HTML Page To display UniversityMark sheet Using Tables Tag
3. Make HTML page as shown below.



StudentMail Sign Up Form

Enrollment
Name
Address
Birth Date (dd/mm/yyyy)
Gender ● Male ○ Female
Hobby □ Music □ Sports
Level 1 level
Email
Password
Confirm Password
Contact

4. Apply the CSS for the Practical 2 University Mark Sheet HTML Page.Use Proper Border, Color, Font, Margin, Logo.
5. Study of Bootstrap. Create a web page using Bootstrap
6. **JavaScript**
   - **a.** Write a JavaScript to find the factorial of a given Number.
   - **b.** Write a JavaScript to find the Fibonacci series up to given number.
   - **c.** Write a javascript for form validation for blank input entry.

   Eg.if u have one textbox to take the input from the user and onebutton submit and if u left the textbox blank without entering any data and click submit button then it should display the massage that enter the data.

d. Perform Validation for Registration Form for Practical 3.

7. **DHTML**
   - **a.** Write a program that changes the Size of an Image when you mouse over the Image and when mouse out image will come in original size.
   - **b.** Write a program for ImageViewer. In that user can navigate through first to last image with next and previous navigation.
   - **c.** Write a program that change the color of background as you mouse over on particular color block and also display the color name in StatusBar
8. **PHP**
   - **a.** Write a PHP script to upload a file to server.
   - **b.** Write a PHP script to get the value from one form and display it into other form.
   - **c.** Write a PHP script for login authentication. Design an html form which takesusername and password from user and validate against stored username andpassword in file.
9. **Write PHP script for storing and retrieving user information from MySql table.**
   - Design an Html page which takes Name,Address,Email and phone fromuser. (registration.php)
   - Store this data in MySql database.
   - Next page displays all user in html table using PHP (display.php)
10. **Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.**
11. **Create REST API using PHP to send the form data like user name, email into the database and test that API.**

Supplementary Learning Materials:

https://developer.mozilla.org/en-US/docs/Web/HTML

https://www.w3schools.com/html/

https://www.tutorialspoint.com/html/index.htm

Course Articulation Matrix:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO 13	PSO 14	PSO 15
1	1	-	1	-	-	-	-	-	-	-	-	-	3	2	2
2	-	3	3	-	-	-	-	-	2	-	-	2	2	1	2
3	-	3	3	2	3	-	1	1	2	-	-	2	3	2	2
4	-	3	3	3	2	1	-	-	2	2	-	2	3	2	2
5	-	1	3	2	2	-	2	-	3	2	3	-	3	2	3
6	-	2	2	1	2	-	1	1	1	-	-	-	3	2	2

1. Slight (Low) 2. Moderate (Medium) 3. Substantial (High) '–' No correlation

****

Detailed Syllabus

Course code:	**CP-5**
Name of the course:	**Mini-project**
Semester:	**6**
Category of Course:	**CP**

Course objectives:

The main objective of Mini Project is to let the students apply the knowledge of theoretical concepts which they have learnt as a part of the curriculum of the minor degree using real time problems or situations.

Teaching & Examination Scheme:

Teaching Scheme			Credits	Examination Marks(Maximum/Passing)		Total
				External		
L	T	P	C	ESE(T)	ESE(P)	
0	0	4	2	0	100/50	100/50

ESE (T): End Semester Examination (Theory)

ESE (P): End Semester Examination (Practical)

Guidelines:

- The mini-project is desirable to be done in a group of 2 students. Each group has to prepare a title related to any engineering discipline, and the title must emulate any real-world problem.
- Submit an early proposal. This proposal is a 1-2page(s) report, describes what the project is about and the final product's output. The project proposal will be submitted to the respective guide

Course Outcomes (CO):

- Understand, plan and execute a Mini Project with team.
- Students will be able to practice acquired knowledge within the chosen area of technology for project development.
- Identify, discuss and justify the technical aspects of the chosen project with a comprehensive and systematic approach.
- Communicate and report effectively project related activities and findings.

****

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

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.

PROGRAM SPECIFIC OUTCOMES (PSOs)

1. Ability to conceptualize interdisciplinary domain knowledge to specific branch of engineering.
2. Ability to acquire employability skillsand deep knowledgein emerging and multidisciplinary areas.
3. Carryout engineering projects in broad areas of engineering.

****