Course code	Course name	Course credit
CMSPCOR01T	Advanced Computer Architecture	4
CMSPCOR02T	Advanced Database Management Systems	4
CMSPCOR03T	Advanced Data Structures and Algorithms	4
CMSPCOR04T	Computer Networks	4
CMSPCOR05P	Data Structures and Algorithms Laboratory	4
CMSPAEC01P	Programming with SQL	2
CMSPCOR06T	Object Oriented Programming	4
CMSPCOR07T	Advanced Operating System	4
CMSPCOR08T	Computer Graphics and Multimedia	4
CMSPCOR09T	Formal Language and Automata Theory	4
CMSPCOR10T	OBJECT ORIENTED Programming Laboratory	4

OutCome

The Student should be able to 1. Learn pipelining concepts with a prior knowledge of stored program methods, 2. Learn about memory hierarchy and mapping techniques. 3. Study of parallel architecture and interconnection network .

The Student should be able to 1. Explore the basic concepts of database systems. 2. Write SQL queries for a given scenario. 3. Describe relational database theory, and be able to write relational algebra expressions for queries. 4. Design logical data models 5. Evaluate and optimize queries 6. Implement transaction processing and concurrency control 7. Develop Object oriented dB, Distributed dB using XML, data warehousing

The Student should be able to 1. Ability in using the appropriate algorithm for searching, sorting, indexing operations 2. Designing of new algorithms 3. Analyzing complexity issues of algorithms

The Student should be able to 1. Describe various components and categories of data communications, types of connections, topologies, protocols and standards, various transmission media and modems. 2. Detect and correct the errors using various algorithmic techniques, be aware of the various Ethernet standards and bridges. 3. Explain various switching techniques used and implement the various routing and router protocols. 4. Illustrate multiplexing and demultiplexing, UDP, TCP protocols and Congestion Control mechanisms. 5. Illustrate Network Applications. 6. Security over cryptography.

At the end of the course the student should be able to: 1. Implement Divide and Conquer algorithm design technique for various applications 2. Implement dynamic programming algorithm design technique for various applications 3. Implement Greedy algorithm design technique for various applications 4. Implement backtracking algorithm design technique for various applications

At the end of the course the student should be able to: 1. Populate and query a database using SQL DML/DDL commands 2. Write programs using PL/SQL including stored procedures , cursors, packages etc. 3. Construct real time database application using current techniques

The student should be able to

1. Explain concepts in object oriented programming.

2. Write simple programs in JAVA.

3. Demonstrate the concept of functions, operator overloading, inheritance through JAVA programs.

4. Demonstrate the concepts of exception handling, generic functions, and templates.

The student should be able to

1) The students will understand the design approaches of advanced operating systems

2) Analyze the design issues of distributed operating systems.

3) Evaluate design issues of multi processor operating systems.

4) Identify the requirements of operating systems.

5) Formulate the solutions to schedule the real time applications.

The Student should be able to

1. Develop algorithms to draw fundamental drawings

2. Develop real-time rendering graphics

3. Create 2D and 3D images

The Student should be able to

1. Apply the theoretical concepts and techniques in designing finite automata

2. Convert regular expressions to FA and minimize Automata.

3. Write context free Grammar and design PDA for the Grammar.

4. Design turing machine and identify recursively enumerable languages.

5. Define undecidability and identify class P and NP problems.

The students should be able to

1. Use platform independent programming for different application.

2. Use multithread and synchronization they can run real time programming.

3. Using applet they can design different APPs.

4. They can handle web page for different networking interface

The students should be able to
1 Make use of algorithms to draw 2D and 3D objects
2 Show transformations and projections for 2D and 3D objects
3 Manipulate a graphical object using clipping algorithms and viewing technique
4 Use an image editing tool for image manipulation and enhancement
5 Utilize the authoring tool to develop a 3D scene and to perform 2D animation
The student should be able to
1. Apply the concepts of life cycle models to choose the appropriate model.
2. Analyse the requirements and design the software.
3. Construct or implement the software based on the industry standards

The student should be able to
1. Differentiate the various phases of a compiler.
2. Apply parsing techniques and able to write Context Free Grammars for various languages.
3. Design the structure of intermediate code for various types of statements and expressions.
4. Design code generator and apply code optimization techniques.
5. Can design own compiler of any work specific application.
The Student should be able to
1. Ability to identify and formulate appropriate AI methods for solving a problem
2. Ability to implement AI algorithms
3. Ability to compare different AI algorithms in terms of design issues, computational complexity, and assumptions

The Student should be able to
1.Explain the software engineering process and project management
2 Demonstrate software requirements and analysis using UML
3 Outline the software design process and user interface
4 Compare and contrast various software testing
5 Discuss about the software integration and project management
6.Use testing tools to do software testing.
7.Use version control tools and create build files
The students should be able to
1. Face any interviews.
2. Interdisciplinary knowledge .
3. Research and development.
4. Exposure to real world.

The Student should be able to

1. Ability to design and model a system
2. Ability to plan and execute well defined objective
3. Ability to work in team at component level and system level
4. Ability to troubleshoot
5. Ability to reuse- or integrate with- existing components
6. Ability to derive performance metrics and assess quantitatively the performance of system
7. Ability to report and present the findings in standard formats