Computer Engineering Syllabus Semester - 2

Topics : Computer engineering
Written on March 06, 2024

1. **Data Structures and Algorithms:** This course covers advanced data structures (e.g., arrays, linked lists, stacks, queues, trees, graphs) and algorithm design techniques (e.g., sorting, searching, recursion, dynamic programming). Students learn how to analyze the time and space complexity of algorithms and apply them to solve computational problems efficiently.

2. **Discrete Mathematics:** This course explores mathematical concepts that are fundamental to computer science and engineering, including sets, relations, functions, logic, proof techniques, graph theory, and combinatorics. Students develop problem-solving skills and mathematical reasoning abilities relevant to computer science applications.

3. **Object-Oriented Programming:** Building on the programming fundamentals learned in the first semester, this course focuses on object-oriented programming principles and practices. Students learn concepts such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction, and apply them to design and implement object-oriented software solutions.

4. **Computer Networks:** This course introduces the principles and protocols of computer networks, including the OSI model, TCP/IP protocol suite, network topologies, routing, switching, LANs, WANs, and network security. Students learn how data is transmitted, routed, and managed in computer networks and gain an understanding of network design and administration.

5. **Database Management Systems:** This course covers the principles and techniques of database management systems (DBMS), including data modeling, relational database design, SQL programming, transaction management, concurrency control, and database administration. Students learn how to design, implement, and query databases to store and retrieve structured data efficiently.

6. **Operating Systems:** This course explores the concepts and components of operating systems, including process management, memory management, file systems, device management, and security. Students learn how operating systems control and coordinate hardware resources to provide a seamless execution environment for software applications.

7. **Digital Electronics:** Building on the basics of digital logic design, this course covers advanced topics such as sequential circuits, flip-flops, counters, registers, memory devices,

programmable logic devices (PLDs), and synchronous/asynchronous sequential circuits. Students learn how to design and analyze complex digital systems using hardware description languages and simulation tools.

8. **Computer Graphics:** This course introduces the principles and techniques of computer graphics, including 2D and 3D graphics programming, raster and vector graphics, geometric transformations, rendering techniques, and graphical user interface (GUI) design. Students learn how to create interactive and visually appealing graphics applications.

9. **Probability and Statistics:** This course provides a foundation in probability theory and statistical methods relevant to computer engineering applications. Topics may include probability distributions, random variables, statistical inference, hypothesis testing, regression analysis, and probabilistic modeling.

10. **Professional Ethics and Human Values:** This course examines ethical issues and social responsibilities in the practice of computer engineering, including professional codes of conduct, intellectual property rights, privacy, security, and ethical decision-making. Students develop ethical awareness and critical thinking skills essential for responsible engineering practice.

11. **Computer Laboratory-II:** This course provides hands-on experience with software development tools, programming languages, and development environments relevant to computer engineering. Students work on practical projects and assignments to reinforce theoretical concepts and gain practical skills in software development and debugging.

12. **Project Work (Mini-Project):** Students work on a mini-project under the guidance of faculty members, applying the knowledge and skills acquired in coursework to solve a real-world problem or explore a specific topic of interest in computer engineering. The project involves requirements analysis, design, implementation, testing, and documentation.