Second Year -Fourth Semester

Branch: Computer Science and Engineering

Course Code	**CS 401**
Course Title	**ANALYSIS AND DESIGN OF ALGORITHMS**
Type of Course	Core
L T P	3 1 3
Credits	4
Course Assessment Methods End Semester Assessment (University Exam.) Continuous Assessment (Sessional, Assignments, Quiz)	 50 50
Course Prerequisites	Introduction to Computer Science and Engineering(CS102), Data Structures (CS301)
Course Objectives (CO)	To understand the different algorithms design techniques and to understand the algorithm analysis approach.
Course Outcome	1. Understand different measures for time and space complexities. 2. Understand the different algorithm design approaches including Divide and Conquer, Greedy, Dynamic Programming, Backtracking and Branch and Bound. 3. Understand P and NP class of problems.

SYLLABUS

***Note for Examiner**- Examiner will set 7 questions of equal marks. First question will cover whole syllabus, having 10 conceptual questions of 1 mark each or 5 questions of 2 mark each and is compulsory. Rest of the paper will be divided into two parts having three questions each and the candidate is required to attempt at least two questions from each part.*

SECTION-A

Introduction:-Revisiting space/time complexity and asymptotic notations; Recurrences: writing recurrences, solving recurrences: iterative substitution, recursion-tree method, Master’s theorem, substitution method.

(8 hours)

Divide and Conquer:- General method,Analysis of divide and conquer based solutions to: Binary Search, Merge sort, Quick sort, Selection sort; finding maximum and minimum using divide and conquer, Strassen’s matrix multiplication.

(7 hours)

Greedy Algorithms:-Elements of Greedy strategy, Activity Selection Problem, Knapsack problem, Single source Shortest paths problem, Minimum Spanning tree problem and analysis of these problems.

(8 hours)

SECTION-B

Dynamic Programming:- Elements of dynamic programming, Assembly-line scheduling problem, Matrix-chain multiplication, Multistage Graph, All Pairs Shortest paths, Longest common subsequence, 0/1 Knapsack.

(12 hours)

Backtracking: - General method, N-Queen's problem, Graph coloring problem, Sum of subsets Problem.

(6 hours)

NP-Completeness:-Polynomial Time, polynomial-time verification, NP-completeness and reducibility, NP-complete problems.

(4 hours)

TEXT BOOKS			
S. No.	**NAME**	**AUTHOR(S)**	**PUBLISHER**
1	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest	Prentice Hall of India
RECOMMENDED BOOKS			
1	Fundamentals of Computer Algorithms	Ellis Horowitz, SartajSahni	Galgotia
2	The Design and Analysis of Computer Algorithms	Aho A.V., Hopcroft J.E., Ullman J.D.	Pearson Education
3	Introduction to the Design and Analysis of Algorithms	Goodman S.E. &Hedetniemi	McGraw-Hill

Branch: Computer Science and Engineering

Course Code	CS 451
Course Title	**ANALYSIS AND DESIGN OF ALGORITHMS (Practical)**
Type of Course	Core
L T P	0 0 3
Credits	1
Course Assessment Methods	
End Semester Assessment	
Continuous Assessment	50

SYLLABUS

Practical should be covered based on the following directions:

1. Divide & Conquer
2. Greedy Method
3. Dynamic Programming
4. Backtracking

Branch: Computer Science and Engineering

Course Code	**CS 402**
Course Title	**WEB TECHNOLOGIES**
Type of Course	Core
L T P	3 1 0
Credits	4
Course Assessment Methods End Semester Assessment (University Exam.) Continuous Assessment (Sessional, Assignments, Quiz)	 50 50
Course Prerequisites	Introduction to Computer Science and Engineering(CS102), Programming Fundamentals(CS101/201)
Course Objectives (CO)	Aim of this course is to familiarize the students with current technologies used in Web development and maintenance 1. To introduce the concepts of Internet ,WWW and underlying technologies 2. To enable the student to use of HTML, DHTML, CSS for Static Webpage creation. 3. To introduce the concept of JavaScript for Client Side programming. 4. To introduce the concept of byte code and Java Programming to develop architecture (Platform) neutral application. 5. To study the concept of XML for data interchange across different platforms 6. To introduce the concept of PHP for various web servers 7. To demonstrate the concept of integrating AJAX and PHP with MySQL.
Course Outcome	1. Understand the core principle on which Internet and WWW operates and ability to create static web pages using HTML, CSS and DHTML. 2. Create dynamic and interactive web contents using the concept of JavaScript, Session and Cookies in Software development. 3. Understand the basic principle of Object Oriented Technology and ability to create powerful but robust standalone application using Java. 4. Understand the concept of add on technologies like AJAX, XML and ability to develop Web Pages using PHP and AJAX and MySQL for server side scripting

SYLLABUS

***Note for Examiner**- Examiner will set 7 questions of equal marks. First question will cover whole syllabus, having 10 conceptual questions of 1 mark each or 5 questions of 2 mark each and is compulsory. Rest of the paper will be divided into two parts having three questions each and the candidate is required to attempt at least two questions from each part.*

SECTION-A

INTERNET AND WORLD WIDE WEB:
Introduction, Internet addressing, ISP, types of Internet connections, introduction to WWW, web browsers, web servers, URL, HTTP, DNS, web applications, tools for web site creation.

(4 hours)

HTML: Introduction to HTML, lists, adding graphics to HTML page, creating tables, linking documents, frames, DHTML and cascading style sheets.

(7 hours)

Java Script: Introduction, programming constructs: variables, operators and expressions, conditional checking, functions and dialog boxes, JavaScript DOM, creating forms, objects like Window, Navigator, History, Location, introduction to cookies,

(11 hours)

SECTION-B

XML: Why XML, XML syntax rules, XML elements, XML attributes, XML DTD displaying XML with CSS.

(6 hours)

PHP: Introduction, syntax, variables, statements, operators, decision making, loops, arrays, strings, forms, get and post methods, functions, cookies, sessions.

(11 hours)

PHP and MySQL: Introduction to MySQL, connecting to MySQL database, creation, insertion, deletion and retrieval of MySQL data using PHP, PHP and XML, XML parsers, XML DOM.

(6 hours)

TEXT BOOKS			
S. No.	**NAME**	**AUTHOR(S)**	**PUBLISHER**
1.	XML How to Program,	Deitel,Deitel,Nieto, and Sandhu	Pearson Education
2.	Java 2: The Complete Reference	Herbert Schieldt	TMH, Fifth Edition
3.	Web Enabled Development Application	Ivan Bayross : Commercial	BPB
4.	HTML,CSS, JavaScript,Perl, Python and PHP	Schafer Textbooks.	Wiley India

Branch: Computer Science and Engineering

Course Code	CS 452
Course Title	**WEB TECHNOLOGIES (Practical)**
Type of Course	Core
L T P	0 0 3
Credits	1
Course Assessment Methods	
End Semester Assessment	
Continuous Assessment	50

SYLLABUS

Practical should be covered based on the following directions:

1. Creation of Web pages using: HTML, DHTML
2. Creation of Web pages using JavaScript
3. Implementing basic concepts of Java
4. Creation of Web pages using AJAX
5. Database and AJAX
6. XML
7. PHP

Branch: Computer Science and Engineering

Course Code	**CS 403**
Course Title	**OPERATING SYSTEM**
Type of Course	Core
L T P	3 1 0
Credits	4
Course Assessment Methods End Semester Assessment (University Exam.) Continuous Assessment (Sessional, Assignments, Quiz)	 50 50
Course Prerequisites	Introduction to Computer Science and Engineering (CS102), Programming Fundamentals (CS101/201), Data Structures (CS301).
Course Objectives (CO)	1. To introduce design and implementation issues of various Operating Systems: batch, multiprogrammed, time sharing, real time, distributed, parallel Operating System structural Components, layered structure, functions 2. To understand concept of processes, CPU Scheduling Algorithms: FCFS, SJF, RR and Priority, Inter Process Communication, Process Synchronization, Critical Sections, Semaphores and Monitors. 3. To introduce Deadlocks Detection , Recovery, Avoidance and Prevention 4. To familiarize with Memory Management using contiguous memory allocation, paging, segmentation, segmentation with paging. 5. To introduce Virtual Memory, demand paging and page replacement algorithms (FIFO, Optimal, LRU), Thrashing. 6. To understand File Systems, directory structure, allocation methods (contiguous, linked, indexed), free-space management (bit vector, linked list, grouping) and Protection mechanisms. 7. To discuss Disk Structure, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN, and LOOK), Disk Management (Disk Formatting, Boot Blocks, and Bad Blocks), Swap Space Management (Swap Space use, Swap Space Location, Swap Space Management). 8. To explore case Studies: Brief introduction of MS-DOS, Windows, UNIX and LINUX.

Course Outcome	1. Design and implement solutions for CPU scheduling, process synchronization and deadlock related problems. 2. Understand the concepts of memory management, Secondary storage management and File system management along with providing solutions for real world problems. 3. Explore features and functionality of MS-DOS, Windows, Unix and Linux.

SYLLABUS

Note for Examiner*- Examiner will set 7 questions of equal marks. First question will cover whole syllabus, having 10 conceptual questions of 1 mark each or 5 questions of 2 mark each and is compulsory. Rest of the paper will be divided into two parts having three questions each and the candidate is required to attempt at least two questions from each part.*

SECTION-A

Introduction: What is an O.S., O.S. Functions; Different types of O.S.: batch, multiprogrammed, time sharing, real time, distributed, parallel; General structure of operating system, O/S services, system calls.

(5 hours)

Process Management: Introduction to processes - Concept of processes, process scheduling, operations on processes; Interprocess Communication, Critical Sections, Mutual Exclusion with Busy Waiting, Sleep and Wakeup, Semaphores, Message passing; CPU scheduling- scheduling criteria, pre-emptive & non-pre-emptive scheduling, Scheduling Algorithms: FCFS, SJF, RR and priority, Threads.

(10 hours)

Deadlocks: Introduction to deadlocks, Conditions for deadlock, Resource allocation graphs, Deadlock Detection and Recovery, Deadlock Avoidance, Deadlock Prevention

(6 hours)

SECTION-B

Memory Management: background, logical vs. physical address space, memory management without swapping; swapping; contiguous memory allocation, paging, segmentation, segmentation with paging; Virtual Memory, demand paging, performance, page replacement, page replacement algorithms (FIFO, Optimal ,LRU); Thrashing.

(6 hours)

File Systems: Files - file concept, file structure, file types, access methods, File attributes, file operations; directory structure, allocation methods (contiguous, linked, indexed), free-space management (bit vector, linked list, grouping), Protection mechanisms.

(6 hours)

Secondary Storage : Disk Structure, Disk Scheduling (FCFS, SSTF, SCAN, C-SCAN, LOOK), Disk Management (Disk Formatting, Boot Blocks, Bad Blocks), Swap Space Management (Swap Space use, Swap Space Location, Swap Space Management)

(6 hours)

Case Studies: Brief introduction of MS-DOS, Windows, UNIX and LINUX.

(6 hours)

TEXT BOOKS			
S. No.	**NAME**	**AUTHOR(S)**	**PUBLISHER**
1.	Operating System Concepts	Silberschatz and Galvin	Addison Wesley Inc.
2	Operating System Design & Implementation	Tanenbaum A.S	Pearson Education.
3	An introduction to Operating Systems Concepts and Practice,	Bhatt and Chandra	Prentice Hall of India Publication

Branch: Computer Science and Engineering

Course Code	CS 453
Course Title	**OPERATING SYSTEM (Practical)**
Type of Course	Core
L T P	0 0 3
Credits	1
Course Assessment Methods End Semester Assessment Continuous Assessment	 50

SYLLABUS

Practical should be covered based on the following directions:

1. Learning Basic Features and Operating Environment of UNIX and LINUX.
2. Introduction to Shell and Shell Commands.
3. Shell programming: creating a script, making a script executable, shell syntax (variables, conditions, control structures, functions, commands).
4. Process: starting new process, replacing a process image, duplicating a process image, waiting for a process.
5. Programming with semaphores.

Branch: Computer Science and Engineering

Course Code	CS 404
Course Title	**SOFTWARE ENGINEERING**
Type of Course	Core
L T P	3 1 0
Credits	4
Course Assessment Methods End Semester Assessment (University Exam.) Continuous Assessment (Sessional, Assignments, Quiz)	 50 50
Course Prerequisites	Introduction to Computer Science and Engineering (CS102), Programming Fundamentals (CS101/201)
Course Objectives (CO)	This course aims to give students a theoretical foundation in software engineering. Students will learn about the principles and methods of software engineering, including current and emerging software engineering practices and support tools. 1. To understand the concept and need of Software Engineering principles, SDLC, process models and tools. 2. To understand project management as an umbrella activity for software development including schedule and cost estimations. 3. To study the concept of software requirements and their changing nature 4. To understand various software architectures and design principle in software Engineering. 5. Understanding good coding practices, including documentation, contracts, regression tests and daily builds. 6. To understand various quality assurance and testing techniques, including unit testing, functional testing and automated testing. 7. To study various CASE tools and understand the methodologies working behind these tools. 8. To understand model based software development using UML.
Course Outcome	1. Demonstrate an understanding of various process models and be able to select appropriate process model for a particular project. 2. Use software cost estimation and scheduling techniques for small programs. 3. Understand SRS and create architecture design for software systems using CASE

	tools. 4. Devise test plan, test case and test suit using Black Box and White Box Testing.

SYLLABUS

***Note for Examiner**- Examiner will set 7 questions of equal marks. First question will cover whole syllabus, having 10 conceptual questions of 1 mark each or 5 questions of 2 mark each and is compulsory. Rest of the paper will be divided into two parts having three questions each and the candidate is required to attempt at least two questions from each part.*

SECTION-A

Introduction:

Introduction to Software Engineering, System Engineering Vs Software Engineering, Software Evolution, Software Characteristics, Cost of Software Production, Software Components, Crisis – Problem and Causes, Challenges in Software Engineering.

(4 hours)

Software Process Models:

SDLC, Waterfall Model, Incremental Model, Prototyping Model, Evolutionary Model, Spiral Model, Rapid Application Development Model, Rational Unified process Model, Agile Methods, Xtreme programming, SEI Capability Maturity Model.

(8 hours)

Software Requirements Analysis and Specification Concepts:

Requirement Engineering, Requirement Elicitation Techniques, Requirements Documentation, Characteristics and Organization of SRS,

(4 hours)

Software Analysis and Design:

Design Principles, Design issues and Approaches, Abstraction, modularity, Coupling, Cohesion, Structured Analysis and Design, DFD, Object oriented Design, Data Design, Architectural design, Interface Design, Component Level Design, Object Oriented Design Concepts, Structured vs. Object Oriented Analysis.

(8 hours)

SECTION-B

Project Management Concepts:

Management Activities, Project Planning, Project Scheduling, Size Estimation – LOC, FP; Cost Estimation Models –COCOMO, COCOMO-II.

(6 hours)

Coding & Testing:

Coding, Coding Standards, Coding Conventions, Programming Style, Verification and Validation, Testing Process, Design of Test Cases, Software Testing Strategies, Unit Testing, Integration Testing, Top Down and Bottom Up Integration Testing, Alpha & Beta Testing, System Testing and Debugging.

(5 hours)

Technical Metrics for Software:

Software Measurements: What and Why, A Framework for Technical Software Metrics, Metrics for the Analysis Model, Metrics for Design Model, Metrics for Source Code, Metrics for Testing, Metrics for Software Quality, Metrics for Maintenance.

(4 hours)

CASE (Computer Aided Software Engineering) and Introduction to UML:

CASE and its Scope, Building blocks of CASE, CASE Tools, CASE Environment, UML Concepts, Use Case Diagrams, Sequence Diagrams, Collaboration Diagrams, Class Diagrams, State Transition Diagrams, Component and Deployment Diagrams.

(6 hours)

<table>
<tr><td colspan="4"><b>TEXT BOOKS</b></td></tr>
<tr><td><b>S. No.</b></td><td><b>NAME</b></td><td><b>AUTHOR(S)</b></td><td><b>PUBLISHER</b></td></tr>
<tr><td>1.</td><td>Software Engineering</td><td>Ian Sommerville</td><td>Pearson Education
Seventh Edition</td></tr>
<tr><td colspan="4"><b>RECOMMENDED BOOKS</b></td></tr>
<tr><td>1</td><td>Software Engineering: A Practitioner's Approach</td><td>R.S. Pressman</td><td>McGraw Hill.
Sixth Edition</td></tr>
<tr><td>2</td><td>Software Engineering: Theory and Practice, ,</td><td>Pfleeger, J.M. Atlee</td><td>Pearson Education
Second Edition</td></tr>
<tr><td>3</td><td>Software Engineering for Students.</td><td>Douglas Bell</td><td>PearsonEducation
Fourth Edition</td></tr>
<tr><td>4</td><td>An Integrated Approach to Software Engineering,</td><td>Pankaj Jalote</td><td>Narosa
Second Edition</td></tr>
<tr><td>5</td><td>Software Engineering</td><td>K.K.Aggarwal, Yogesh Singh</td><td>New Age International.
Second Edition</td></tr>
</table>

Branch: Computer Science and Engineering

Course Code	CS 454
Course Title	**SOFTWARE ENGINEERING (Practical)**
Type of Course	Core
L T P	0 0 3
Credits	1
Course Assessment Methods End Semester Assessment Continuous Assessment	50

SYLLABUS

Practical should be covered based on the following directions:

1. Study the features of MS-Project.
2. Use MS-Project/OpenProj/similar tool to draft project plan for a particular project case study.
3. Use MS-Project/OpenProj/similar tool to generate various reports like Gantt chart, Network diagram, Resource usage sheet.
4. Use MS-Project/OpenProj/similar tool to track the progress of a project.
5. Study the concepts of UML modeling.
6. Use Rational Rose/StarUML/similar tool to generate use case diagrams.
7. Use Rational Rose/StarUML/similar too to generate sequence diagrams.
8. Use Rational Rose/StarUML/similar too to generate class diagrams.
9. Use Rational Rose/StarUML/similar too to generate collaboration diagrams.
10. Study the features of a particular CASE tool for requirements specification, analysis, design and cost estimation.
11. Apply each of the above tools to a particular case study.

Branch: Computer Science and Engineering

Course Code	CS 405
Course Title	**COMPUTER ARCHITECTURE & ORGANIZATION**
Type of Course	Core
L T P	3 1 0
Credits	4
Course Assessment Methods End Semester Assessment (University Exam.) Continuous Assessment (Sessional, Assignments, Quiz)	 50 50
Course Prerequisites	Introduction to Computer Science and Engineering (CS102) Microprocessors (CS 304)
Course Objectives (CO)	This course offers a good understanding of the various functional units of a computer system and prepares the student to be in a position to design a basic computer system.
Course Outcome	1. Understand basic organization and functional units of Computer and principles and implementation of computer arithmetic. 2. Understand and Design the different functional units and control unit of CPU 3. Understand the concepts of memory and I/O organization 4. Understand pipelining and parallel processing concepts. 5. Evaluate and design of a basic computer system

SYLLABUS

***Note for Examiner**- Examiner will set 7 questions of equal marks. First question will cover whole syllabus, having 10 conceptual questions of 1 mark each or 5 questions of 2 mark each and is compulsory. Rest of the paper will be divided into two parts having three questions each and the candidate is required to attempt at least two questions from each part.*

SECTION-A

Basic organization of computers, Block level description of the functional units as related to the execution of a program; Fetch, decode and execute cycle.

(6 hours)

Machine instructions, Instruction set architectures, Assembly language programming, addressing modes, instruction cycles, registers and storage, addressing modes; discussions about RISC versus CISC architectures; Inside a CPU

(6 hours)

Information representation, Floating point representation (IEEE 754), computer arithmetic and their implementation; Fixed-Point Arithmetic: Addition, Subtraction, Multiplication and Division, Arithmetic Logic Units control and data path, data path components, design of ALU and data path, controller design; Hardwired and Micro programmed Control .

(10 hours)

SECTION-B

Memory Technology, static and dynamic memory, Random Access and Serial Access Memories, Cache memory and Memory Hierarchy, Address Mapping, Cache updation schemes, Virtual memory and memory management unit.

(8 hours)

I/O subsystems: Input-Output devices such as Disk, CD-ROM, Printer etc.; Interfacing with IO devices, keyboard and display interfaces; Basic concepts Bus Control, Read Write operations, Programmed IO, Concept of handshaking, Polled and Interrupt-driven I/O, DMA data transfer.

(8 hours)

Pipeline Processing, Instruction and Arithmetic Pipeline, Pipeline hazards and their resolution, Parallel Processing.

(7 hours)

TEXT BOOKS			
S. No.	**NAME**	**AUTHOR(S)**	**PUBLISHER**
1.	Computer Organization	V. Carl Hamacher, Safwat G. Zaky and Zvonko G. Vranesic	Tata McGraw-Hill series (2002)
2	Computer Organization and Design	David Patterson and John Hennessey	Elsevier (2008)
3.	Computer System Architecture	M. Morris Mano	Pearson Third Edition
4.	Computer Architecture and Organization	J.P. Hayes	Tata McGraw-Hill Third Edition
5.	Computer Organization and Architecture	William Stallings	Pearson Seventh Edition