B.Tech. *Computer Engineering*-SEMESTER III												
Course Code	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
CECSC04	CC	Web Technology	3	0	2	4	15	15	40	15	15	Computer Programming and Data Structure

COURSE OUTCOMES

1. To understand the development and transition of the web.
2. To Learn creating the web pages and apply the styles
3. To learn the web programming for simple day to day work.
4. To learn fetching the object using latest technologies and using them to process information
5. To write a full-fledged web based application and deploy it.

COURSE CONTENT

UNIT-1

Introduction , Web Browser , Web 2.0

Introduction , W3C,Web 2.0 , Personal, Distributed and Client/Server Computing , Browser Portability , Software Technologies , Web Resources, Customizing Browser Setting ,Searching the Internet, Keeping Track of Your Favorite Sites, File Transfer Protocol (FTP),Online Help, Web Resources, Web 2.0?, Search, Content Networks , User-Generated Content, Blogging, Social Networking, Social Media, Tagging , Social Bookmarking , Software Development , Rich Internet Applications (RIAs), Web Services, Mashups, Widgets and Gadgets, Location-Based Services, XML, RSS, Atom, JOSN and VolP, Web 2.0 Monetization Models, Web 2.0 Business Models, Future of the Web , Where to GO for more Web 2.0 Information ,

XHTML , Cascading Style Sheets(CSS)

Introduction , Editing XHTML, First XHTML Example, W3CXHTML Validation Service, Headings, Linking , Images, Special Characters and Horizontal Rules , Lists, Tables, Forms, Internal Linking , Meta Elements, Inline Styles, Embedded Style Sheers, Conflicting Style, Linking External Style sheers, Positioning Elements, Backgrounds, Element Dimensions, Box Model and Text Flow, Media Types, Building a CSS Drop-Down Menu, User Style Sheets, CSS 3, Web Resources

UNIT-2

JavaScript: Introduction to Scripting

Introduction , Simple Program: Displaying a Line of Text in a Web Page , Modifying Our First Program , Obtaining User Input with Prompt Dialogs, Dynamic Welcome Page, Adding Integers, Memory Concepts, Arithmetic, Decision Making: Equality and Relational Operators, Web Resources,

JavaScript: Control Statements

Introduction, Algorithms, Pseudo code, Control Structures, i f Selection Statement, i f …else selection Statement, Formulating Algorithms: Counter-Controlled Repetition , Formulating Algorithms: Sentinel-Controlled Repetition, Formulating Algorithms: Nested Control Statements, Assignment Operators, Increment and Decrement Operators, Essentials of Counter-Controlled Repetition , For Repetition Statement , Examples Using the for Statement, Switch Multiple-Selection Statement , do….while Repetition Statement , break and continue Statements, Labeled break and Continue Statements, Logical Operators, Summary of Structures Programming , Web Resources

JavaScript: Functions

Introduction , Program Modules in JavaScript, Programmer-Definitions Functions, Function Defamations , Random Number Generation, Example: Game of Chance, Another Example: Random Image Generation , JavaScript Global Functions, Recursion , Recursion vs. Iteration ,

JavaScript: Arrays

Introduction, Arrays, Declaring and Allocating Arrays, Examples Using Arrays, Random Image Generator Using Arrays, References and Reference Parameters, Passing Arrays to Functions, Sorting Arrays, Searching Arrays: Linear Search and Binary Search, Multidimensional Arrays, Building an Online Quiz, Introduction to Object Technology, Math Object, String Object, Fundamentals of Characters and Strings, Methods of the string Object , Character-Processing Methods, Searching Methods, Splitting Steins and Obtaining Substrings, XHTML Markup Methods, Date Object, Boolean and Number Objects

Document object Model (DOM): Objects and Collections, introduction , Modeling a Document: DOM Nodes and Trees , Traversing and Modifying a DOM Tree, DOM Collections, Dynamic Styles, Summary of the DOM Objects and Collection, Web Resources , JavaScript: Events, Introduction, Registering Event Handlers, Event onload , Event onmousemove, the event Object and this , Rollovers with onmouseover and onmouseout, From Processing with onfoucus and onblur, More Form Processing with onsubmit and onreset , Event Building , More Events

UNIT-3

XML and RSS

Introduction, XML Basics, Structuring Data, ZXML Namespaces, Document Type Definitions (DTDs), W3CXML Schema Documents, XML Vocabularies, MathML™, Other Markup Languages, Extensible Stylesheet Language and XSL Transformations, Document Object Model (DOM), RSS,

Ajax- Enabled Rich Internet Applications

Introduction, Traditional Web Application vs. Ajax Applications, Rich Internet Applications (RIAs) with Ajax, History of Ajax, “Raw” Ajax Example Using the XMLHttpRequest Object, Using XML and the DOM, Creating a Full-Scale Ajax-Enabled Application, Dojo Toolkit

UNIT-4

IIS and Apache: introduction, architecture, client – server side scripting, requesting documents.

Database: Introduction, RDBMS, SQL – simple queries, ADO.NET object model, Java DB/ Apache Derby

PHP: introduction, basics, form processing and business logic, connecting to a database, using cookies.

Ruby on Rails: Intro, Ruby, Rails Framework, scripting

UNIT-5

ASP.NET and ASP.NET Ajax: introduction, creating and running a simple web form example, relationship, generating XHTML code, web controls, sessions tracking.

JavaServer Faces Web Applications:
Java Web technologies, creating and running a simple application in Netbeans, examining a JSP file, event processing life cycle, JSF components, Text vs. graphics components, validation, session cookies, web services

Guidelines for Project work:
Exercises based on these technologies

REFERENCE BOOKS

1. Deitel and Deitel: Internet and Worldwide Web programming, Pearson
2. Frank Barbier: Reactive Internet programming, ACM Books
3. Tara Calishain: Google hacks, O'Reilly Media
4. Sergei DunaevAdvanced Internet Programming, IT Master

B.Tech. *Computer Engineering*-SEMESTER III												
Course Code	**Type**	**Subject**	**L**	**T**	**P**	**Credits**	**CA**	**MS**	**ES**	**CA**	**ES**	**Pre-requisites**
CECSC05	**CC**	**Database Management Systems**	**3**	**0**	**2**	**4**	**15**	**15**	**40**	**15**	**15**	

COURSE OUTCOMES

At the end of the course students will be able to

CO1: understand fundamentals of database management systems.

CO 2: design database models and learn database languages to write queries to extract information from databases.

CO 3: Identify database anomalies and improve the design of database management system

CO 4: understand transaction management and concurrency control.

CO 5: understand storage organization and database recovery.

COURSE CONTENT

UNIT 1

Introduction: Database management system Characteristics of the Database, Database Systems and Architecture, Data Models, Schemes & Instances, DBMS Architecture & Data Independence, Database administrator & Database Users, Database Languages & Interfaces, DDL, DML, DCL, Overview Relational Data Base Management Systems

UNIT 2

Data Modeling: Data modeling using The Entity-Relationship Model – Entities, Attributes and Relationships, Cardinality of Relationships, Strong and Weak Entity Sets, Generalization, Specialization, and Aggregation, Translating your ER Model into Relational Model, Relationships of higher degree.

UNIT 3

Relational Model, Languages & Systems: Relational Data Model concepts, Relational Model Constraints, integrity constraints ,Keys domain constraints, referential integrity, assertions triggers, foreign key
Relational Algebra and calculus, SQL. Database security.

Relational Data Base Design: Functional Dependencies & Normalization for Relational Databases, Functional Dependencies, Normal Forms Based on Primary Keys, (1NF, 2NF, 3NF & BCNF), Lossless Join and Dependency Preserving Decomposition, Functional dependencies and its closure, covers and equivalence.

UNIT 4

Transaction Management: Transaction Concept and State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability: Testing of serializability, Serializability of schedules, conflict & view serializable schedule.
Concurrency Control Techniques: Lock-Based Protocols, Timestamp-based Protocols, validation based protocol. Deadlock Handling

UNIT 5

Recovery System

Recoverability: Failure Classification, Storage Structure, Recovery and Atomicity, Log-based Recovery, Shadow Paging, Recovery with Concurrent Transactions

Storage organization : Indexing, Hashing ,file storage.

List of Experiments:

Following is only a suggestive list of experiments. For better coverage faculty may increase the list of experiments.

Q 1: Consider the following relational schema

SAILORS (sid, sname, rating, date_of_birth)
BOATS (bid, bname, color)
RESERVES (sid, bid, date, time slot)

Write the following queries in SQL and relational algebra

a) Find sailors who've reserved at least one boat
b) Find names of sailors who've reserved a red or a green boat in the month of March.
c) Find names of sailors who've reserved a red and a green boat
d) Find sid of sailors who have not reserved a boat after Jan 2018.
e) Find sailors whose rating is greater than that of all the sailors named "John"
f) Find sailors who've reserved all boats
g) Find name and age of the oldest sailor(s)
h) Find the age of the youngest sailor for each rating with at least 2 such sailors

Q2. Consider the following relational schema:

CUSTOMER (cust_num, cust_lname , cust_fname, cust_balance);
PRODUCT (prod_num, prod_name, price)
INVOICE (inv_num, prod_num, cust_num, inv_date ,unit_sold, inv_amount);

Write SQL queries and relational algebraic expression for the following

a) Find the names of the customer who have purchased no item. Set default value of Cust_balance as 0 for such customers.
b) Write the trigger to update the CUST_BALANCE in the CUSTOMER table when a new invoice record is entered for the customer.
c) Find the customers who have purchased more than three units of a product on a day.
d) Write a query to illustrate Left Outer, Right Outer and Full Outer Join.
e) Count number of products sold on each date.
f) As soon as customer balance becomes greater than Rs. 100,000, copy the customer_num in new table called "GOLD_CUSTOMER"
g) Add a new attribute CUST_DOB in customer table

Q 3: Consider the following relational schema

DEPARTMENT(Department_ID, Name, Location_ID)
JOB (Job_ID , Function)
EMPLOYEE (Employee_ID, name, DOB, Job_ID , Manager_ID, Hire_Date, Salary, department_id)

Answer the following queries using SQL and relational algebra:

a) Write a query to count number of employees who joined in March 2015
b) Display the Nth highest salary drawing employee details.

c) Find the budget (total salary) of each department.
d) Find the department with maximum budget.
e) Create a view to show number of employees working in Delhi and update it automatically when the database is modified.
f) Write a trigger to ensure that no employee of age less than 25 can be inserted in the database.

Q4: PROJECT

Students are required to develop a DBMS for the applications assigned to them. Following items are required to be submitted for the project

a) Problem Statement
b) ER model/ Relational Model
c) Integrity Constraints implemented
d) Suitable Queries to create and manage database

Note: Students have to make sure that they have defined proper integrity constraints to ensure consistency of database used in assignments as well as project.

SUGGESTED READINGS:

Text book:

1. Korth ,Silbertz, Sudarshan, ”Data base concepts”, McGraw-Hill, 2013

Reference books

1.Elmasri, Navathe, “Fundamentals of Database systems”, Addison Wesley, 2010
2.Date C.J., ”An Introduction to Database systems”, Addison-Wesley Longman, Inc.,2004

B.Tech. *Computer Engineering*-**SEMESTER III**												
Course Code	**Type**	**Subject**	**L**	**T**	**P**	**Credits**	**Evaluation Scheme (Percentage weights)**					**Pre-requisites**
							Theory			**Practical**		
							CA	**MS**	**ES**	**CA**	**ES**	
CECSC06	**CC**	**Design and Analysis of Algorithms**	**3**	**0**	**2**	**4**	**15**	**15**	**40**	**15**	**15**	**Programming Data Structures**

COURSE OUTCOMES

1. To be able to analyze the asymptotic performance of algorithms.
2. To be able to write rigorous correctness proofs for algorithms.
3. Ably demonstrate a familiarity with major algorithms and data structures.
4. To be able to apply important algorithmic design paradigms and methods of analysis.
5. To be able to synthesize efficient algorithms in common engineering design situations.

CONTENTS

UNIT I

Design and Analysis of Algorithms Asymptotic notations and their significance, introduction to RAM model of computation, complexity analysis of algorithms, worst case and average case. Basic introduction to algorithmic paradigms like divide and conquer, recursion, greedy, etc.

UNIT II

Searching: binary search trees, balanced binary search trees, AVL trees and red-black trees, B-trees, skip lists, hashing Priority queues, heaps, Interval trees, tries. Order statistics. Sorting: comparison based sorting - quick sort, heap sort, merge sort: worst and average case analysis. Decision tree model and (worst case) lower bound on sorting. Sorting in linear time - radix sort, bucket sort, counting sort, etc. String matching.

UNIT III

Graph Algorithms: BFS, DFS, connected components, topological sort, minimum spanning trees, shortest paths - single source and all pairs.Models of computation: RAM model and its logarithmic cost. Formal introduction to algorithmic paradigms: divide and conquer, recursion, dynamic programming, greedy, branch and bound, etc. Advanced data structures: Fibonacci heap, union-find, splay trees. Amortized complexity analysis

UNIT IV

Randomized algorithms: Randomized algorithms to be introduced a bit early, i.e., before NP-completeness to highlight randomization as an algorithmic technique. Application areas: Geometric algorithms: convex hulls, nearest neighbor, Voronoi diagram, etc. Algebraic and number-theoretic algorithms: FFT, primality testing, etc.

UNIT V

Graph algorithms: network flows, matching, etc. Optimization techniques: linear programming Reducibility between problems and NP-completeness: discussion of different NP-complete problems like satisfiability, clique, vertex cover, independent set, Hamiltonian cycle, TSP, knapsack, set cover, bin packing, etc. Backtracking, branch and bound, Approximation algorithms: Constant ratio approximation algorithms.

REFERENCE BOOKS

1. E. Horowitz, S. Sahni, and S. Rajsekaran, "Funadmentals of Computer Algorithms," Galotia Publication
2. T .H . Cormen, C .E .Leiserson, R .L .Rivest "Introduction to Algorithms", PHI.
3. Sedgewich, Algorithms in C, Galgotia
4. Berman. Paul, "Algorithms, Cengage Learning".
5. Richard Neopolitan, Kumar SS Naimipour, "Foundations of Algorithms"

B.Tech. *Computer Engineering*-**SEMESTER III**

Course Code	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
CECSC07	**CC**	**Computer Architecture and Organization**	**3**	**1**	**0**	**4**	**25**	**25**	**50**	**-**	**-**	**Computer Programming and Data Structure**

COURSE OUTCOMES

1. To understand the architecture of modern processors and organization of its components, and relationship between hardware and software in digital machines.
2. To design instructions and corresponding logic circuits for a simple CPU with its essential components such as ALU, a register file, memory and input-output.
3. To understand the organization of computer systems
4. To understand the computation standards and using them in writing algorithms
5. To appreciate the evolving technology that governs the evolution of modern computers and continue to keep abreast of state-of-art in computing technology

COURSE CONTENT

UNIT-1

Overview of computer organization: Characteristics of a general purpose computer, The stored program concept, von Neumann architecture, Harvard architecture, Programmer's model - the Instruction set architecture (ISA), ISA design and performance criteria, Basic computer organization with CPU, memory and IO subsystems, Interconnect busses, Evolution of CISC and RISC based processors and their merging.

UNIT-2

Instruction Set Architectures: Machine instruction, Machine cycle and Instruction cycles, Instruction Set: memory and non-memory reference instructions, instruction categories: data movement, data manipulation, program control and machine control instructions, CISC types addressing modes and instruction formats, RISC type addressing modes and instruction formats.

UNIT-3

Central Processing Unit: Specification of a simple CPU using RTL, Design of the data path for the simple CPU, Designing the hardwired control path for the simple CPU, Performance analysis of the simple CPU, Enhancement of the ISA for the simple CPU and design extensions, Characteristics of RISC CPU design: ISA characteristics, pipelining, data and instruction caches, Practical case studies in CISC type and RISC type CPU designs.

UNIT-4

Microprogrammed Control Unit: Control memory system, Microinstruction-sequencing, conditional branch, mapping and subroutines, direct, horizontal and vertical microcoding, micro-instruction format and symbolic representation, design of micro-control unit for a simple CPU, applications of microprogramming

Memory organization: Memory hierarchy, Cache organization: Direct, associative and Set associative cache, Auxiliary memory organization, RAID organizations
Input output organization: IO interfacing, Asynchronous data transfer, Programmed IO, Interrupt driven IO, Priority schemes, Direct Memory Access, Serial communication techniques

UNIT-5

Computer arithmetic: Design of Binary addition and subtraction units, Algorithms for multiplication and division and their implementation, Floating point arithmetic, etc.
Pipelined architecture: Basic concepts of pipelining, Speedup and throughput, Minimum Average Latency, Instruction pipeline.
GPU architecture: Hardware Basics, Execution Model, GPU instruction set architecture, NVIDIA GPU instruction set architecture

Guidelines for Project work:

- Exercises using assembly-level programming and debugging to illustrate the working of instructions in the ISA of a CISC based /RISC based processor. These exercises should illustrate the status of various registers, flags, counters and pointers after data

movement, data manipulation, program control, and stack operations.
- Semester-long group project on the design and simulation /hardware emulation of a simple processor.

REFERENCE BOOKS

1. William stallings, “Computer Organization and Architectue, PHI” 2. M. Morris Mano,
2. M. Morris Mano, “Computer System Architecture” , PHI
3. J.D. Carpinelli, “Computer Systems Organization and Architecture,” Pearson Education
4. Heuring and Jordan, Pearson Education, “Computer Systems Design and Architecture”
5. Tor M. Aamodt, Wilson Wai Lun Fung, Timothy G. Rogers General-Purpose Graphics Processor Architectures

B.Tech. *Computer Engineering*-**SEMESTER III**

Course Code	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
CEECC08	CC	Microprocessors and Microcontrollers	3	0	2	4	15	15	40	15	15	None

COURSE OUTCOMES

1. Acquire knowledge of architecture and programming of microprocessors.
2. Understand the salient features of the x86 architecture.
3. Acquire hands-on knowledge of interfacing microprocessors with peripherals.
4. Understand the architecture and working of microcontrollers and their utility.
5. Acquire introductory knowledge about high-end microprocessors and microcontrollers.

COURSE CONTENT

Unit 1 - Intel 8085 microprocessor: Basic concepts of microprocessor, microcomputer, microcontroller. Architecture (pins, signals, buses, register set), addressing modes, instruction set (instruction format, opcode, mnemonic), subroutines, timing diagrams and t-states of different instructions, programming, recursive programs, vectored and non-vectored interrupts and interrupt handling of 8085.

Unit 2 - Intel 8086 microprocessor: Architecture (pins, bus interface unit, execution unit, register set, pipelining), memory addressing, segmentation, instruction set (data transfer, arithmetic, logic, string, long and short control transfer and processor control), timing diagrams, operating modes, programming, assemblers, address-objects, parameter passing to subroutines, hardware and software interrupts and interrupt handling of 8086.

Unit 3 - Interfacing of microprocessors: Interfacing a microprocessor with RAM and ROM chips, address allocation and decoding techniques. Interfacing with LED, LCD, ADC, DAC, toggle switch and keypad. Memory-mapped i/o. Interfacing with 8255 programmable peripheral interface (architecture, ports, i/o modes and BSR mode). Basic architecture and features of 8254 programmable timer, 8257 programmable DMA controller, 8259 programmable interrupt controller, 8279 programmable keyboard and display controller and 8087 math coprocessor.

Unit 4 - Microcontrollers: 8051 microcontroller: architecture, i/o ports, memory organization, addressing modes, instruction set, simple programs. Introduction to IoT: basic architecture, sensing and actuating, application domains.

Unit 5 - High-end microprocessors and microcontrollers: Important features of 32-bit processors, RISC and Pentium. Implementation of memory management schemes like segmentation, paging and virtual memory at the hardware level. Introduction to Arduino: basic architecture, hardware and software, simple programs.

Guidelines for practical work:

1. Write an assembly program to generate the numbers of the Fibonacci series.
2. Write an assembly program to clear all flags without using any data transfer instruction.
3. Write an assembly program to search for a number in a list.
4. Write an assembly program to sort a list.
5. Write an assembly program to copy a list from one part of the memory to another.
6. Write an assembly program to multiply two numbers using successive additions.
7. Write an assembly program to calculate the square root of a number.
8. Write an assembly program to calculate the factorial of a number using recursion.
9. Write a self-replicating assembly program.
10. Interface 8255 with a microprocessor and use all its modes.
11. Interface 8254 with a microprocessor and use it to generate differ types of clock signals.
12. Interface 8259 with a microprocessor and use all its features.
13. Interface 8257 with a microprocessor and write a program to control a keypad and a LED display.
14. Design digital systems with Arduino and simple sensors and actuators.

SUGGESTED READINGS

1. Ramesh S. Gaonkar, “Microprocessor Architecture, Programming, and Applications with the 8085” Prentice Hall.
2. D. V. Hall, “Microprocessor and Interfacing Programming & Hardware” TMH – 2nd Edition.
3. S. P. Morse, “8086 Primer: An Introduction to Its Architecture, System Design and Programming” Hayden Book Co.
4. S. Monk, “Programming Arduino: Getting Started with Sketches”, 2nd Edition, McGraw-Hill.

5. M.A. Mazidi et. al. “The 8051 Microcontroller and Embedded Systems: Using Assembly and C” Pearson Publishers.