



**JSS COLLEGE OF ARTS, COMMERCE AND
SCIENCE**

(Autonomous)

B N ROAD, MYSURU- 570 025

**DEPARTMENT OF COMPUTER
SCIENCE**

Syllabus

CHOICE BASED CREDIT SYSTEM

For B.Sc PROGRAMME

Physics, Mathematics, Computer Science

2017-18

PROGRAMME: BSc PMCs, PROGRAMME CODE: BSc-02 (2017-18)

UG-Computer Science - CBCS Scheme

Year	Sem	Corse Code	Title	Hours / Week		Credits	Maximum Marks			Exam Duration	Total Mark
				L	T/ P		IA		Exam		
				L	T/ P	L:T:P	C1	C2			
I Year	I	CMA 25002	DSC 3A - Computer Science – I								
			Problem Solving Using Computer	4	4	4:0:2	15	15	70	3 Hours	100
I Year	II	CMB 25002	DSC 3B - Computer Science – II								
			Computer System Organization and Architecture	4	4	4:0:2	15	15	70	3 Hours	100
II Year	III	CMC 25002	DSC 3C - Computer Science – III								
			Data Structures and File Processing	4	4	4:0:2	15	15	70	3 Hours	100
II Year	IV	CMD 25002	DSC 3D - Computer Science – IV								
			Operating Systems	4	4	4:0:2	15	15	70	3 Hours	100
III Year	V		DSE 3 - Computer Science – V (Choose Any One)								
		CME 25002	A - Database Management Systems	4	0	4:0:1	15	15	70	3 Hours	100
		CME2 5202	B - Computer Networks	4	0	4:0:1	15	15	70	3 Hours	100
		CME 25402	C - Software Engineering	4	0	4:0:1	15	15	70	3 Hours	100
			SEC 3# - Computer Application(Choose Any One)								
		CME 38002	A - Office Automation	0	2	0:0:1	15	15	70	2 Hours	100
		CME 38202	B - XML Programming	0	2	0:0:1	15	15	70	2 Hours	100
		CME 38402	C - R Programming	0	2	0:0:1	15	15	70	2 Hours	100

introducing this paper is only after Approvals of AC / GB

Year	Sem	Corse Code	Title	Hours / Week		Credits	Maximum Marks			Exam Duration	Total Mark
				L	T/ P		IA		Exam		
				L	T/ P	L:T:P	C1	C2			

UG-Computer Science - CBCS Scheme

III Year	VI		DSE 6 - Computer Science –VI(Choose Any One)								
		CMF 25002	A - Internet Technology	4	0	4:0:1	15	15	70	3 Hours	100
		CMF 25202	B - Mobile Applications	4	0	4:0:1	15	15	70	3 Hours	100
		CMF 25402	C - Cloud Computing	4	0	4:0:1	15	15	70	3 Hours	100
			SEC 4 - Computer Science –VII(Choose Any One)								
		CMF 38002	A - Android Programming	1	2	1:0:1	15	15	70	2 Hours	100
		CMF 38202	B - PHP Programming	1	2	1:0:1	15	15	70	2 Hours	100
		CMF 38602	C - System Administration & Maintenance	1	2	1:0:1	15	15	70	2 Hours	100

UG-Computer Science - CBCS Scheme

Assessment Maximum marks - 100

Course type	C1		C2		C3 Exam Marks		Assigned Marks (Percentage)			Total
	Theory	La b	Theory	La b	Theory	Lab	Theory	Lab	IA	
DSC	10	05	10	05	70	70	50	20	30	100
DSE	10	05	10	05	70	70	50	20	30	100
DSE (non practical)	15	-	15	-	70	-	70	--	30	100
SEC	15	-	15	-	50	-	70	--	30	100

Note:

1. C1 will be conducted for 20 Marks (Theory) with onehour duration - 10 Marks (Lab) with continuous assessment and it will be reduced to assigned marks.
2. C2 will be conducted for 20 Marks (Theory) with onehour duration - 10 Marks (Lab) with continuous assessment and it will be reduced to assigned marks.
3. C3 will be conducted for 70 Marks (Theory) with three hours duration - 70 Marks (Lab) with 3 hours duration and to be reduced to assigned marks.
4. For non-practical course C3 will be conducted for 70 Marks (Theory) with three hours duration.
5. In case of SEC, C1 and C2 will be conducted for 15 Marks each with one hour duration and C3 will be conducted for 50 Marks with 2 hours duration.

UG-Computer Science - CBCS Scheme

Programme Outcome for Bachelor of Science in Physics, Mathematics, Computer Science:

After completing the graduation in the Bachelor of Science the students are able to:

- PO1. Demonstrate proficiency in Mathematics and the Mathematical concepts needed for a proper understanding of Physics.
- PO2. Demonstrate the ability to justify and explain their thinking and/or approach.
- PO3. Develop state-of-the-art laboratory and professional communication skills.
- PO4. Apply the scientific method to design, execute, and analyze an experiment
- PO5. Explain scientific procedures and experimental observations.
- PO6. Understand the value of Mathematical proof
- PO7. Demonstrate proficiency in writing and understanding proofs.
- PO8. Apply mathematical problems and solutions in aspects of science and technology.
- PO9. Gain experience to investigate the real world problems
- PO10. Apply mathematical ideas and models to those problems.
- PO11. Apply Mathematical principles for computing and logical design.
- PO12. Design, implement, and evaluate a computational system to meet desired needs within realistic constraints.
- PO13. Use the System principles in the design and development of software for systems of varying complexity.

UG-Computer Science - CBCS Scheme

Programme Specific Outcome

Bachelor of Science in Physics, Mathematics, Computer Science

After completing the graduation in the Bachelor of Science the students are able to:

- PSO1. Find career opportunities
- PSO2. Develop competence to write competitive examinations.
- PSO3. Develop proficiency in the analysis of complex physical problems
- PSO4. Use of mathematical or other appropriate techniques to solve problems
- PSO5. Create a hypothesis and appreciate how it relates to broader theories.
- PSO6. Demonstrate skills in the use of Computers
- PSO7. Join as Entry level Technical job role for an IT Industry
- PSO8. Build small database ERP software/ web applications.

SEMESTER I

Course code: CMA25002

DSC 3A: Computer Science-I

Problem Solving Using Computer

Credits: Theory - 04, Practical - 02

Theories: 60 Lectures

COURSE OUTCOME:

After successful completion of the course, the student is able to

- CO1. Deliberate in depth Computer Fundamentals
- CO2. Understand in details with features Computer Organization
- CO3. Learn the details of Problem Solving Techniques
- CO4. Deliberate the characteristics of Programming Languages
- CO5. Understand in details with examples - Python Programming Languages
- CO6. Specify in depth OOPs, Event Driven and GUI features in Python

Unit - 1

(15Lectures)

Computer Fundamentals: Introduction to Computers: Characteristics of Computers, Uses of computers, Types and generations of Computers.

Basic Computer Organization: Units of a computer, CPU, ALU, memory hierarchy, registers, I/O devices.

Techniques of Problem Solving: Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation, Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.

Unit - 2

(15 Lectures)

Overview of Programming: Structure of a Python Program, Elements of Python

Introduction to Python: Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator). Input and Output Statements, Control statements (Looping-while Loop, for Loop, Loop Control)

UG-Computer Science - CBCS Scheme

Unit - 3

(15 Lectures)

Creating Python Programs: Conditional Statement- if...else, Difference between break, continue and pass.

Structures: Numbers, Strings, Lists, Tuples, Dictionary, Date & Time, Modules, Defining Functions, Exit function, default arguments.

Unit - 4

(15 Lectures)

Introduction to Advanced Python: Objects and Classes, Inheritance, Regular Expressions, Event Driven Programming, GUI Programming.

Reference:

1. P. K. Sinha&PritiSinha , “Computer Fundamentals”, BPB Publications, 2007.
2. Dr. Anita Goel, Computer Fundamentals, Pearson Education, 2010.
3. T. Budd, Exploring Python, TMH, 1st Ed, 2011
4. Python Tutorial/Documentation www.python.org 2010
5. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online.2012

Problem Solving Using ComputerLab

Software Lab using Python

Section: A (Simple programs)

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user’s choice.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria
Grade A: Percentage ≥ 80
Grade B: Percentage ≥ 70 and < 80
Grade C: Percentage ≥ 60 and < 70
Grade D: Percentage ≥ 40 and < 60
Grade E: Percentage < 40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number

UG-Computer Science - CBCS Scheme

6. WAP to find sum of the following series for n terms: $1 - 2/2! + 3/3! - \dots - n/n!$
7. WAP to calculate the sum and product of two compatible matrices.

Section: B (Visual Python)

All the programs should be written using user defined functions, wherever possible.

1. Write a menu-driven program to create mathematical 3D objects
 - I. Curve
 - II. Sphere
 - III. Cone
 - IV. Arrow
 - V. Ring
 - VI. Cylinder.
2. WAP to read n integers and display them as a histogram.
3. WAP to display sine, cosine, polynomial and exponential curves.
4. WAP to plot a graph of people with pulse rate p vs. height h. The values of p and h are to be entered by the user.
5. WAP to calculate the mass m in a chemical reaction. The mass m (in gms) disintegrates according to the formula $m=60/(t:2)$, where t is the time in hours. Sketch a graph for t vs. m, where $t \geq 0$.
6. A population of 1000 bacteria is introduced into a nutrient medium. The population p grows as follows:
$$P(t) = (15000(1:t))/(15: e)$$
Where the time t is measured in hours. WAP to determine the size of the population at given time t and plot a graph for P vs t for the specified time interval.
7. Input initial velocity and acceleration, and plot the following graphs depicting equations of motion:
 - I. Velocity wrt time ($v=u:at$)
 - II. Distance wrt time ($s=u*t:0.5*a*t*t$)
 - III. Distance wrt velocity ($s=(v*v-u*u)/2*a$)

SEMESTER II

Course code: CMB25002

DSC 3B: Computer Science-II

Computer System Organization and Architecture

Credits: Theory - 04, Practical - 02

Theories: 60 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Deliberate in details with examples Boolean algebra and logic circuits
- CO2. Learn the details of Data Representation and Computer Arithmetic
- CO3. Learn in depth Computer Organization and Design
- CO4. Learn the details of architecture of CPU
- CO5. Deliberate the classification and characteristics of Basic Computer Programming Concepts
- CO6. Write down in depth Basic Computer Programming Concepts
- CO7. Learn the classification and characteristics of Input -Output organization

Unit - 1

(15 Lectures)

Introduction: Logic gates, Boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexors, registers, counters and memory units.

Data Representation and basic Computer Arithmetic: Number systems, complements.

Unit - 2

(15 Lectures)

Fixed and floating point representation, character representation, addition, subtraction, magnitude comparison.

Basic Computer Organization and Design: Computer registers, bus system, instruction set.

UG-Computer Science - CBCS Scheme

Unit - 3

(15 Lectures)

timing and control, instruction cycle, memory reference, input-output and interrupt.

Central Processing Unit: Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control.

Unit - 4

(15 Lectures)

Programming the Basic Computer: Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming.

Input-output Organization: Peripheral devices, I/O interface, Modes of data transfer, direct memory access.

Reference:

1. M. Mano, Computer System Architecture, Pearson Education 1992.
2. A. J. Dos Reis, Assembly Language and Computer Architecture using C:: and JAVA, Course Technology, 2004
3. W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Prentice Hall of India ,2009
4. Digital Design, M.M. Mano, Pearson Education Asia, 1979

Computer System Organization and Architecture Lab

1. Verification of Basic gates(AND,OR,NOT)
2. Verification of Universal gates(NAND,NOR,EX-OR)
3. Verification of NAND gate as a Universal gate
4. Verification of NOR gate as a Universal gate
5. Verification of Demorgan's theorem
6. Verification of Half adder & Full Adder
7. Verification of Half subtractor & Full Subtractor
8. Verification of Half adder & Half subtractor using NAND gate
9. Conversion of Binary to Gray & gray to Binary Code
10. Simplification of Boolean Expressions
11. Simplification of Boolean Expressions using K-Map
12. Flip-Flops:SR FF(clock, without clock)
13. JK FF
14. Toggle FF
15. Delay FF
16. Multiplexer
17. De-multiplexer
18. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.
 - a. ADD f. BSA
 - b. AND g. ISZ
 - c. LDA
 - d. STA
 - e. BUN
19. Simulate the machine for the memory-reference instructions referred in above question with I= 1 and address part = 082. The instruction to be stored at address 026 in RAM. Initialize the memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.
20. Modify the machine created in Practical 1 according to the following instruction format:

Instruction format

0	2	3	4	15
Opcode	I		Address	

- a. The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit Address. There are only two addressing modes, I = 0 (direct addressing) and I = 1(indirect addressing).
- b. Create a new register I of 1 bit.
- C. Create two new microinstructions as follows:**
 - i. Check the opcode of instruction to determine type of instruction (Memory

UG-Computer Science - CBCS Scheme

- Reference/Register Reference/Input-Output) and then jump accordingly.
- ii. Check the I bit to determine the addressing mode and then jump accordingly.

SEMESTER III

Course code: CMC25002

DSC 3C:Computer Science-III

Data Structures and File Processing

Credits: Theory - 04, Practical - 02
60 Lectures

Theories:

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Learn the classification, characteristics and understanding of Data structures
- CO2. Specify the details of Searching Techniques
- CO3. Deliberate in details with examples Basic Concepts of Memory Management Techniques
- CO4. Understand in depth File System Operations
- CO5. Specify the characteristics of File Organization Methods
- CO6. Deliberate in details with examples of Storage Devices

Unit - 1

(15 Lectures)

Basic Data Structures:Introduction, Abstract data structures- stacks, queues, linked lists and binary trees. **Sets:**Dictionary implementation, use of priority queues, hashing, binary trees, balanced trees, setswith merge-find operations.

Unit - 2

(15 Lectures)

Searching: Internal and external searching, use of hashing and balancing techniques.

Memory Management: Garbage collection algorithms for equal sized blocks, storage allocationfor objects with mixed size, buddy systems.

Unit - 3

(15 Lectures)

Physical Devices: Characteristics of storage devices such as disks and tapes, I/O buffering.

BasicFile System Operations: Create, open, close, extend, delete, read-block, write-block, protectionmechanisms.

Unit - 4

(15 Lectures)

UG-Computer Science - CBCS Scheme

File Organizations: Sequential, indexed sequential, direct, inverted, multi-list, directory systems, Indexing using B-tree, B+ tree and their variants, hashing – hash function, collision handling methods, extendible hashing.

Reference:

1. M.T. Goodrich, R. Tamassia and D. Mount, Data Structures and Algorithms in C++, John Wiley and Sons, Inc., 2004.
2. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, Introduction to Algorithms, 2nd Ed., Prentice-Hall of India, 2006.
3. Robert L. Kruse and A.J. Ryba, Data Structures and Program Design in C++, Prentice Hall, Inc., NJ, 1998.
4. B. Stroustrup, The C++ Programming Language, Addison Wesley, 2004
5. D.E. Knuth, Fundamental Algorithms (Vol. I), Addison Wesley, 1997

Data Structures and File Processing Lab

Part - A

1. Program to find lower triangular and upper triangular matrices for the given matrix.
2. Write an interactive program to insert an element at the given position and delete an element at the specified position in the given array.

UG-Computer Science - CBCS Scheme

3. Program to search an element identify the number of occurrences with locations in linear array.
4. Program to sort the given M x N matrix row-wise and column-wise using bubble sorting technique.
5. Write an interactive program to search an element in the given linear array using linear and binary searching technique.
6. Write a program to Merge two sorted arrays.

Part – B

7. Write an interactive program to implement the following operations on stack using arrays
 - a.PUSH
 - b.POP
8. Program to implement Tower of Hanoi problem.
9. Write an interactive program to perform insertion and deletion operations in Linear Queue using arrays.
10. Write an interactive program to perform insertion and deletion operations in Circular Queue using arrays.
11. Write an interactive program to insert a node in a linked list at the front, delete a node from the rear and display.
12. Write an interactive program to implement preorder, post order and in order traversal of a binary tree using linked list.

SEMESTER IV

Course code: CMD25002

DSC 3D - Computer Science – IV

Operating Systems

Credits: Theory - 04, Practical - 02

Theories: 60 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Learn in details with examples system software
- CO2. Learn the details of Operating System organization
- CO3. Understand the classification and characteristics of Process Management and Scheduling mechanisms
- CO4. Understand in depth Memory Management and allocation strategies
- CO5. Learn in details with examples basic concepts of shell scripting
- CO6. Understand in depth basic Linux environment

Unit - 1

(15 Lectures)

Introduction: System Software, Resource Abstraction, OS strategies. Types of operating systems - Multiprogramming, Batch, Time Sharing, Single user and Multiuser, Process Control & Real Time Systems

Operating System Organization: Factors in operating system design, basic OS functions, implementation consideration; process modes, methods of requesting system services – system calls and system programs.

Unit - 2

(15 Lectures)

Process Management: System view of the process and resources, initiating the OS, process address space, process abstraction, resource abstraction, process hierarchy, Thread model

Unit - 3

(15 Lectures)

Scheduling: Scheduling Mechanisms, Strategy selection, non-pre-emptive and pre-emptive strategies.

Memory Management: Mapping address space to memory space.

Unit - 4

(15 Lectures)

Memory allocation strategies, fixed partition, variable partition, paging, virtual memory

Shell introduction and Shell Scripting: What is shell and various type of shell, Various editors present in Linux, Different modes of operation in vi editor, What is shell script, Writing and executing the shell script, Shell variable (user defined and system variables), System calls, Using system calls, Pipes and Filters, Decision making in Shell Scripts (If else, switch), Loops in shell, Functions, Utility programs (cut, paste, join, tr, uniq utilities), Pattern matching utility (grep)

Reference:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
4. W. Stallings, Operating Systems, Internals & Design Principles , 5th Edition, Prentice Hall of India. 2008.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

Operating Systems Lab

Software Lab based on Operating Systems

Note: Following exercises can be performed using Linux or Unix

1. Usage of following commands: ls, pwd, tty, cat, who, who am I, rm, mkdir, rmdir, touch, cd.
2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands: chmod, grep, tput (clear, highlight), bc.
4. Write a shell script to check if the number entered at the command line is prime or not.

UG-Computer Science - CBCS Scheme

5. Write a shell script to modify “cal” command to display calendars of the specified months.
6. Write a shell script to modify “cal” command to display calendars of the specified range of months.
7. Write a shell script to accept a login name. If not a valid login name display message – “Entered login name is invalid”.
8. Write a shell script to display date in the mm/dd/yy format.
9. Write a shell script to display on the screen sorted output of “who” command along with the total number of users .
10. Write a shell script to display the multiplication table any number,
11. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
12. Write a shell script to find the sum of digits of a given number.
13. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
14. Write a shell script to find the LCD(least common divisor) of two numbers.
15. Write a shell script to perform the tasks of basic calculator.
16. Write a shell script to find the power of a given number.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.
20. Program to show the pyramid of special character “*”.

SEMESTER V

Course code: CME25002

DSE 3A: Elective: Computer Science – V

Database Management Systems

Credits: Theory - 04, Practical - 01

Theory: 60 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Understand the characteristics of DBMS with examples
- CO2. Deliberate the details of types of database languages with examples
- CO3. Learn the details of ER- Diagrams and Relationship
- CO4. Understand in depth Basic concepts of Relational Model
- CO5. Learn in details with examples MYSQL Commands
- CO6. Learn in details with examples in PL-SQL

Unit - 1

(15 Lectures)

Introduction to Database Management Systems: Definition of Data, Information, DBMS, Data base system application, Purpose of database systems, Characteristics of DB – Self describing nature, Insulation between programs, data and data Abstraction (data Independence), support of multiple views of the data, sharing of data and multiples transaction processing, Storage management, Database language – DDL, DML,DCL.

File processing system v/s DBMS, Data models, Levels of Abstraction in a DBMS, Three Schema architecture,Characteristics of database approach,, data models,DBMS architecture and data independence.

Unit - 2

(15 Lectures)

Entity Relationship and Enhanced ER Modeling: Entity types,Entity Sets, Attributes, and Keys, Relationships, Relationship Types, Roles, and Structural Constraints,Weak Entity Types, ER Diagrams, Naming Conventions ,SQL99:Schema Definition, constraints, and object modeling

UG-Computer Science - CBCS Scheme

Unit - 3

(15 Lectures)

Relational Data Model: Basic concepts, Relational Constraints and Relational Database Schemas, Update Operations and Dealing with Constraint Violations, Basic Relational Algebra Operations.

Database design: ER and EER to relational mapping, functional dependencies, normal forms-first normal form, second normal forms, third normal form BCNF

Unit - 4

(15 Lectures)

MYSQL (SQL/PL-SQL) : SQL VS. SQL * PLUS: SQL COMMANDS AND DATA TYPES, OPERATORS AND EXPRESSIONS, INTRODUCTION TO SQL * PLUS.

MANAGING TABLES AND DATA:

- CREATING AND ALTERING TABLES (INCLUDING CONSTRAINTS)
- DATA MANIPULATION COMMAND LIKE INSERT, UPDATE, DELETE
- SELECT STATEMENT WITH WHERE, GROUP BY AND HAVING, ORDER BY, DISTINCT, SPECIAL OPERATOR LIKE IN, ANY, ALL BETWEEN, EXISTS, LIKE
- JOIN, BUILT IN FUNCTIONS OTHER DATABASE OBJECTS
- VIEW • SYNONYMS, INDEX TRANSACTION CONTROL STATEMENTS
- COMMIT, ROLLBACK, SAVEPOINT INTRODUCTION TO PL/SQL
- SQL V/S PL/SQL • PL/SQL BLOCK STRUCTURE
- LANGUAGE CONSTRUCT OF PL/SQL (VARIABLES, BASIC AND COMPOSITE DATA TYPE, CONDITIONS LOOPING ETC.)
- % TYPE AND % ROWTYPE
- USING CURSOR (IMPLICIT, EXPLICIT)

Reference:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishnan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6th Edition, McGraw Hill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

Database Management Systems Lab

Software Lab based on Database Management Systems

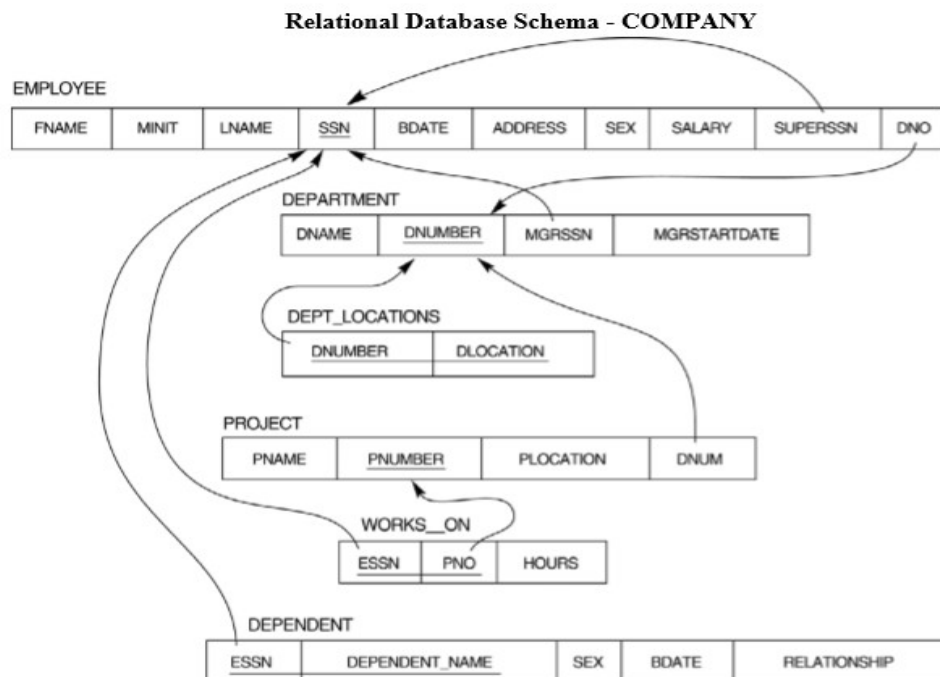
The following concepts must be introduced to the students: **Note:** MS Access/MySQL may be used.

DDL Commands

- Create table, alter table, drop table

DML Commands

- Select, update, delete, insert statements
- Condition specification using Boolean and comparison operators (and, or, not, =, <, >, <=, >=, <=)
- Arithmetic operators and aggregate functions (Count, sum, avg, Min, Max)
- Multiple table queries (join on different and same tables) • Nested select statements
- Set manipulation using (any, in, contains, all, not in, not contains, exists, not exists, union, intersect, minus, etc.)
- Categorization using group by.....having
- Arranging using order by



1. Create tables with relevant foreign key constraints
2. Populate the tables with data
3. Perform the following queries on the database :

UG-Computer Science - CBCS Scheme

- a. Display all the details of all employees working in the company.
 - b. Displayssn, lname, fname, address of employees who work in department no 7.
 - c. Retrieve the birthdate and address of the employee whose name is 'Franklin T. Wong'
4. Retrieve the name and salary of every employee
 5. Retrieve all distinct salary values
 6. Retrieve all employee names whose address is in 'Bellaire'
 7. Retrieve all employees who were born during the 1950s
 8. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive)
 9. Retrieve the names of all employees who do not have supervisors
 10. Retrieve SSN and department name for all employees
 11. Retrieve the name and address of all employees who work for the 'Research' department
 12. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate.
 13. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.
 14. Retrieve all combinations of Employee Name and Department Name
 15. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project.
 16. Increase the salary of all employees working on the 'ProductX' project by 15%.
Retrieve employee name and increased salary of these employees.
 17. Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employee first name.
 18. Select the names of employees whose salary does not match with salary of any employee in department.
 19. Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee.
 20. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford.
 21. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with proper headings.
 22. Find the sum of the salaries and number of employees of all employees of the 'Marketing' department, as well as the maximum salary, the minimum salary, and the average salary in this department
 23. Select the names of employees whose salary is greater than the average salary of all employees in department 10.
 24. For each department, retrieve the department number, the number of employees in the department, and their average salary.
 25. For each project, retrieve the project number, the project name, and the number of employees who work on that project.

UG-Computer Science - CBCS Scheme

26. Change the location and controlling department number for all projects having more than 5 employees to 'Bellaire' and 6 respectively.
27. For each department having more than 10 employees, retrieve the department no, no of employees drawing more than 40,000 as salary.
28. Insert a record in Project table which violates referential integrity constraint with respect to Department number. Now remove the violation by making necessary insertion in the Department table.
29. Delete all dependents of employee whose ssn is '123456789'.
30. Delete an employee from Employee table with ssn = '12345'(make sure that this employee has some dependents, is working on some project, is a manager of some department and is supervising some employees). Check and display the cascading effect on Dependent and Works on table. In Department table MGRSSN should be set to default value and in Employee table SUPERSSN should be set to NULL
31. Perform a query using alter command to drop/add field and a constraint in Employee table.

SEMESTER V

Course code: CME25202

**DSE3B: Elective:Computer Science - V
Computer Networks**

Credits: Theory - 04, Practical - 01

Theory: 60 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Understand the Elements of Data Communications and network Systems
- CO2. Learn in depth Transmission Media
- CO3. Understand in details with examples Network Models
- CO4. Understanding the various classifications and characteristics of Protocols
- CO5. Learn in depth Error Detection and Corrections Algorithms
- CO6. Learn in detail of Network Security

Unit - 1

(15 Lectures)

Basic concepts: Components of data communication, standards and organizations, Network Classification, Network Topologies ; network protocol; layered network architecture; overview of OSI reference model; overview of TCP/IP protocol suite.

Unit - 2

(15 Lectures)

Physical Layer: Cabling, Network Interface Card, Transmission Media Devices- Repeater, Hub, Bridge, Switch, Router, Gateway.

Data Link Layer: Framing techniques; Error Control; Flow Control Protocols; Shared media protocols - CSMA/CD and CSMA/CA.

Unit - 3

(15 Lectures)

UG-Computer Science - CBCS Scheme

Network Layer: Virtual Circuits and Datagram approach, IP addressing methods – Subnetting; Routing Algorithms (adaptive and non-adaptive)

Transport Layer: Transport services, Transport Layer protocol of TCP and UDP

Unit - 4 **(15 Lectures)**

Application Layer: Application layer protocols and services – Domain name system, HTTP, WWW, telnet, FTP, SMTP

Network Security: Common Terms, Firewalls, Virtual Private Networks.

Reference:

1. B.A. Forouzan: Data Communication and Networking, 4th Edition, Tata McGraw Hill, 2007.
2. D.E. Comer, Internetworking with TCP/IP, Vol. I, Prentice Hall of India, 1998.
3. W. Stalling, Data & Computer Communication, 8th edition, Prentice Hall of India, 2006.
4. D. Bertsekas, R. Gallager, Data Networks, 2nd edition, Prentice Hall of India, 1992.

Computer Networks Lab

Software Lab based on Computer Networks:

Implement the concepts of Computer Networks such as:

1. Simulate Checksum Algorithm.
2. Simulate CRC Algorithm
3. Simulate Stop & Wait Protocol.
4. Simulate Go-Back-N Protocol.
5. Simulate Selective Repeat Protocol.and so on....

SEMESTER V

Course code: CME25402

DSE 3C: Elective: Computer Science - V Software Engineering

Credits: Theory - 04, Practical - 01

Theory: 60 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Understand in details with examples Concepts of Software process
- CO2. Specify the details of Software requirements and analysis
- CO3. Learn in depth Design concepts and principles of software engineering
- CO4. Understand in depth software Configuration Management and Project Management
- CO5. Learn in details with examples Software Testing
- CO6. Specify in depth trends in software engineering

Unit - 1

(15 Lectures)

Software Process: Introduction ,S/W Engineering Paradigm , life cycle models (water fall, incremental, spiral, evolutionary, prototyping, object oriented) , System engineering, computer based system, verification, validation, life cycle process, development process, system engineering hierarchy.

Software requirements: Functional and non-functional, user, system, requirement engineering process, feasibility studies, requirements, elicitation, validation and management, software prototyping, prototyping in the software process, rapid prototyping techniques, user interface prototyping, S/W document.

Unit - 2

(15 Lectures)

Analysis and modeling, data, functional and behavioral models, structured analysis and data dictionary.

Design Concepts and Principles: Design process and concepts, modular design, design heuristic, design model and document, Architectural design, software architecture, data design, architectural design, transform and transaction mapping, user interface design, user interface design principles. Real time systems, Real time

UG-Computer Science - CBCS Scheme

software design, system design, real time executives, data acquisition system, monitoring and control system.

Unit - 3

(15 Lectures)

Software Configuration Management: The SCM process, Version control, Change control, Configuration audit, SCM standards.

Software Project Management: Measures and measurements, S/W complexity and science measure, size measure, data and logic structure measure, information flow measure. Estimations for Software Projects, Empirical Estimation Models, Project Scheduling.

Unit - 4

(15 Lectures)

Testing: Taxonomy of software testing, levels, test activities, types of s/w test, black box testing, testing boundary conditions, structural testing, test coverage criteria based on data flow, mechanisms, regression testing, testing in the large. S/W testing strategies, strategic approach and issues, unit testing, integration testing, validation testing, system testing and debugging.

Trends in Software Engineering: Reverse Engineering and Re-engineering – wrappers – Case Study of CASE tools.

Reference:

1. Roger S.Pressman, Software engineering- A practitioner's Approach, McGraw-Hill
2. Ian Sommerville, Software engineering, Pearson education Asia, 6th edition, 2000.
3. PankajJalote- An Integrated Approach to Software Engineering, Springer Verlag, 1997.
4. James F Peters and WitoldPedryez, "Software Engineering – An Engineering Approach", John Wiley and Sons, New Delhi, 2000.
5. Ali Behforooz and Frederick J Hudson, "Software Engineering Fundamentals", Oxfor University Press, New Delhi, 1996.
6. Pfleeger, "Software Engineering", Pearson Education India, New Delhi, 1999.
Carlo Ghezzi, Mehdi Jazayari and Dino Mandrioli, "Fundamentals of Software Engineering", Prentice Hall of India, New Delhi, 1991.

Software Engineering Lab

Lab based on Software Engineering

1. Practical Title
 - Problem Statement,
 - Process Model
2. Requirement Analysis
 - Creating a Data Flow
 - Data Dictionary,
 - Use Cases
3. Project Management
 - Computing FP
 - Effort
 - Schedule, Risk Table, Timeline chart
- 1 Design Engineering
 - Architectural Design
 - Data Design, Component Level Design
5. Testing
 - Basis Path Testing

Sample Projects

- DTC Route Information: Online information about the bus routes and their frequency and fares
- Car Pooling: To maintain a web based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
- Patient Appointment and Prescription Management System
- Organized Retail Shopping Management Software
- Parking Allocation System
- Wholesale Management System

SEMESTER V

Course code: CME38002

**SEC 4A: Elective: Computer Application (Practical)
Office Automation**

**Credits: Theory - 00,
Practical - 01**

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Understand the details of fundamentals of Computer
- CO2. Learn in depth Hardware and Software
- CO3. Learn the details of Computer Peripherals
- CO4. Understand the details of Programming Languages
- CO5. Deliberate in details with examples office automation Tools
- CO6. Deliberate in depth Operating System and the User Interface
- CO7. Understand in details of Internet and its usages

Practical List for WORD:

1. Create a **telephone directory**.
 - The heading should be 16-point Arial Font in bold
 - The rest of the document should use 10-point font size
 - Other headings should use 10-point Courier New Font.
 - The footer should show the page number as well as the date last updated.
2. Design a time-table form for your college.
 - The first line should mention the name of the college in 16-point Arial Font and should be bold.
 - The second line should give the course name/teacher's name and the department in 14-point Arial.
 - Leave a gap of 12-points.
 - The rest of the document should use 10-point Times New Roman font.
 - The footer should contain your specifications as the designer and date of creation.
3. Create the following document: A newsletter with a headline and 2 columns in portrait orientation, including at least one image surrounded by text.

Practical List for EXCEL:

1. Create a student worksheet containing roll numbers, names and total marks. Open a document in Word and insert the excel worksheet using:-

UG-Computer Science - CBCS Scheme

- i) Copy/Paste
 - ii) Embedding
 - iii) Linking
2. The term wise marks for APS class of 20 students are stored in 3 separate sheets named term1, term2 and term3. Create 4th worksheet that contains student names and their total and average marks for the entire year. Give proper headings using headers. Make the column headings bold and italic. The 4th worksheet should contain college name as the first line. Make it bold, italic and center it.
3. Consider the following employee worksheet:-

Full Name (First Last)	Grade 1/2/3	Basic Salary	HRA	PF	Gross	Net	(VA)VehicleAllowance

HRA is calculated as follows:

Grade	HRA %(of Basic)
1	40%
2	35%
3	30%

Gross = Basic + HRA + VA

Net = Gross - PF

PF is 8% for all Grades

VA is 15000, 10000 and 7000 for Grades 1, 2 and 3.

- i) Find max, min and average salary of employees in respective Grade
 - ii) Count no. of people where VA>HRA
 - iii) Find out most frequently occurring grade.
 - iv) Extract records where employee name starts with "A" has HRA>10000
 - v) Print Grade wise report of all employees with subtotals of net salary and also grand totals. Use subtotal command.
 - vi) Extract records where Grade is 1 or 2 and salary is between 10000 and 20000 both inclusive.
4. In a meeting of a marketing department of an organization it has been decided that price of selling an item is fixed at Rs40. It was resolved to increase the sell of more of more items and getting the profit of Rs40,000/. Use Goal Seek of find out how many items you will have to sell to meet your profit figure.

5. Consider the following worksheet for APS 1st year students:-

S.No.	Name	PH	CH	BY	MT	CS	Total Marks	%	Grade
1									
2									

Grade is calculated as follows:-

UG-Computer Science - CBCS Scheme

If % ≥ 90 Grade A

If % ≥ 80 & < 90 Grade B

If % ≥ 70 & < 80 Grade C

If % ≥ 60 & < 70 Grade D

Otherwise students will be declared fail.

- i) Calculate Grade using if function
- ii) Sort the data according to total marks
- iii) Apply filter to display the marks of the students having more than 65% marks.
- iv) Draw a pie chart showing % marks scored in each subject by the topper of the class.
- v) Draw the doughnut chart of the data as in (iv)
- vi) Enter the S.No. of a student and find out the Grade of the student using VLOOKUP.
- vii) Extract all records where name
 - a) Begins with "A"
 - b) Contains "A"
 - c) Ends with "A"

Practical List for Power Point:

- 2 Create five Power point slides. Each slide should support different format. In these slides
Explain areas of applications of IT. Make slide transition time as 10 seconds.
- 3 Create five Power Point slides to give advantages/disadvantages of computer, application
of computers and logical structure of computer.
- 4 Create five Power Point slides detailing the process of internal assessment. It should be a
self-running demo.

SEMESTER V

Course code: CME38202

**SEC 4B: Elective: Computer Application (Practical)
Elective:XML Programming**

Credits: Theory - 00, Practical - 01

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Understand the details of Elements of XML Programming
- CO2. Write down in details with examples DTD
- CO3. Deliberate in details with examples XML Schemas
- CO4. Deliberate the characteristics of CSS
- CO5. Learn in details with application CSS
- CO6. Deliberate in details with examples XSL
- CO7. Learn the details of XML Security
- CO8. Learn in details with examples XML and JAVA
- CO9. Learn in details with examples XML and ASP.Net

Software Lab Based on XML:

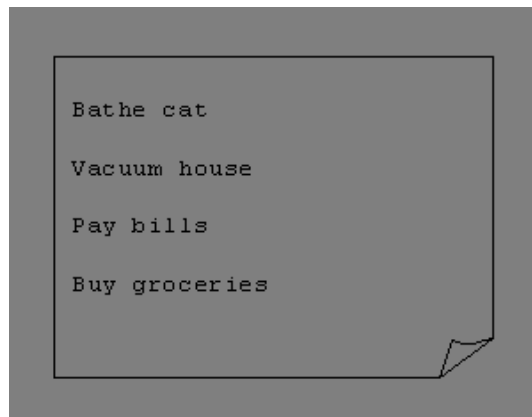
Exercise #1 - Information Structure

In this exercise, student will practice identifying the structure of an information object.

For the sample document provided below:

Label the information structures you see, including containing structures.

12. Draw a tree representation of the structure.



Exercise 2# Deconstructing an XML Document

UG-Computer Science - CBCS Scheme

In this exercise, student will practice identifying the explicit structure within an XML document. In a sense, this is the reverse of what you did in Exercise #1. For the sample XML markup below, create a document-like representation (or a simple drawing) for the content contained within the XML tags:

```
<book>
<coverInfo>
<title>The XML Handbook</title>
<author>Charles F. Goldfarb</author>
<author>Paul Prescod</author>
<edition>Second</edition>
<description>The definitive XML resource: applications, products, and
technologies. Revised and expanded—over 600 new pages.
</description>
</coverInfo>
</book>
```

Exercise #3 - Creating XML Markup

In this exercise, create some XML markup based on the tree representation from Exercise #1 above, and the content from the original sample document.

Exercise #4 - Well-Formedness

This exercise checks your understanding of the constraints for well-formedness. Are the following document instances well-formed? Explain any NO answers.

```
<list><title>The first list</title><item>An item</list>
<item>An item</item><item>Another item</item>
<para>Bathing a cat is a <emph>relatively</emph> easy task as long as
the cat is willing.</para>
<bibl><title>How to Bathe a Cat<author></title>Merlin
Bauer<author></bibl>
```

Exercise #5-Well Formedness

This exercise is a bit more challenging than the previous example. Here is a fragment of an XML document instance. Identify all the places where it fails to match the constraints for well-formedness.

```
<PROCEDURE><TITLE>How to Bathe a Cat</TITLE>
<OVERVIEW>
This procedure tells you how to bathe a cat. <WARNING></OVERVIEW>Cats
don't like to take baths. You could get hurt doing this. Be sure to obtain all
the required protective gear before you start.
</WARNING><EQUIPEMENT><ITEM>Hockey Mask <ITEM>Padded Full-body
Kevlar Armor</ITEM><ITEM>Tub full of warm water</ITEM><ITEM>Towels
</ITEM><ITEM>First Aid kit</ITEM><ITEM>Cat
Shampoo</ITEM><EQUIPEMENT><INSTRUCTIONS><STEP> Locate the cat,
```

UG-Computer Science - CBCS Scheme

who by now is hiding under the bed.</STEP><STEP>Place the cat in the tub of water.</STEP><ITEM>Using the First Aid kit, repair the damage to your head and arms.</STEP><STEP>Place the cat back in the tub and hold it down.</STEP><STEP>Wash it really fast, then make an effort to dry it with the towels.</STEP><STEP>Decide not to do this again.</STEP></INSTRUCTIONS>

Note: Cover more exercises based on XML Programming theory concepts.

SEMESTER V

Course code: CME38402

SEC 4C: Elective: Computer Application (Practical) R Programming

Credits: Theory - 00, Practical - 01

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Learn in depth Elements of R Programming
- CO2. Understand the details of R programming Packages
- CO3. Understand in details with examples of Data Management
- CO4. Deliberate the characteristics of Data Management
- CO5. Understand in depth Graphical Procedures
- CO6. Deliberate in depth Portability Distribution
- CO7. Software Lab Based on R Programming

1. Write a program that prints 'Hello World' to the screen.
2. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n
3. Write a program that prints a multiplication table for numbers up to 12.
4. Write a function that returns the largest element in a list.
5. Write a function that computes the running total of a list.
6. Write a function that tests whether a string is a palindrome.
7. Implement the following sorting algorithms: Selection sort, Insertion sort, Bubble Sort
8. Implement linear search.
9. Implement binary search.
10. Implement matrices addition, subtraction and Multiplication

SEMESTER VI

Course code: CMF25002

**DSE6A: Elective:Computer Science - VII
Internet Technologies**

Credits: Theory - 04, Practical - 01

Theory: 60 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Learn the details of HTML tags
- CO2. Understand the details of Basic CSS and implements
- CO3. Understand the details of Basic Concepts of Java Scripts
- CO4. Learn the Core Java Programming
- CO5. Write down in details with application and Usage of JDBC
- CO6. Learn in detail of JSP Environment

Unit - 1

(15 Lectures)

Introduction to Web Design: Introduction to hypertext markup language (html) document type definition, creating web pages, graphical elements, lists, hyperlinks, tables, web forms, inserting images, frames.

Customized Features: Cascading style sheets, (css) for text formatting and other manipulations.

JavaScript: Data types, operators, functions, control structures

Unit - 2

(15 Lectures)

Java Script events and event handling.

Java: Use of Objects, Array and Array List class, Designing classes, Inheritance, Input/Output, Exception Handling.

Unit - 3

(15 Lectures)

JDBC: JDBC Fundamentals, Establishing Connectivity and working with connection interface, Working with statements, Creating and Executing SQL Statements, Working with Result Set Objects.

UG-Computer Science - CBCS Scheme

JSP: Introduction to JavaServer Pages, HTTP and Servlet Basics, The Problem with Servlets, The Anatomy of a JSP Page, JSP Processing, JSP Application Design with MVC

Unit - 4

(15 Lectures)

Setting Up the JSP Environment, Implicit JSP Objects, Conditional Processing, Displaying Values, Using an expression to Set an Attribute, Declaring Variables and Methods, Error Handling and Debugging, Sharing Data Between JSP Pages, Requests, and Users, Database Access.

Reference Books:

1. Web Enabled Commercial Application Development Using Html, Dhtml, javascript, Perl CgiBy Ivan Bayross, BPB Publications, 2009.
2. BIG Java Cay Horstmann, Wiley Publication , 3rd Edition., 2009
3. Java 7 ,The Complete Reference, Herbert Schildt, 8th Edition, 2009.
4. The Complete Reference J2EE, TMH, Jim Keogh, 2002.
5. Java Server Pages, Hans Bergsten, Third Edition, O'Reilly Media December 2003.

Internet TechnologiesLab

Software Lab based on Internet Technologies

JAVA Script

1. Create a student registration form. Create functions to perform the following checks:
 - a. Roll number is a 7-digit numeric value
 - b. Name should be an alphabetical value (String)
 - c. Non-empty fields like DOB
2. Implement a static password protection.
3. Write a java script
 - a. To change the colour of text using SetTimeout()
 - b. To move an image across screen using SetInterval()

JAVA Programs

1. WAP to find the largest of n natural numbers.
2. WAP to find whether a given number is prime or not.
3. WAP to print the sum and product of digits of an Integer and reverse the Integer.
4. Write a program to create an array of 10 integers. Accept values from the user in that array. Input another number from

UG-Computer Science - CBCS Scheme

the user and find out how many numbers are equal to the number passed, how many are greater and how many are less than the number passed.

5. Write java program for the following matrix operations:
 - a. Addition of two matrices
 - b. Summation of two matrices
 - c. Transpose of a matrixInput the elements of matrices from user
6. Write a java program that computes the area of a circle, rectangle and a Cylinder using function overloading.

JDBC

1. Create a table 'Student' and 'Teacher' in 'College' database and insert two rows in this newly created table using JDBC API and do the following:
 - a. Update an already created table 'Teacher' in 'College' database by updating a teacher's name, with "Dr." appended before the name, whose name is "Rita".
 - b. Repeat the same thing for all the teachers using PreparedStatement.
 - c. Delete the student with ID=3 from 'Student' database.
 - d. Insert two students to the ResultSet returned by the query which selects all students with FirstName="Ayush". The database must also get updated along with ResultSet.
2. Create a procedure in MySQL to count the number of Rows in table 'Student'. Use Callable Statement to call this method from Java code.

JSP Practical list

1. Display the pattern:

1

1 2

1 2 3

Take 'n' in a textbox from user. Display this pattern using

- Scriptlets
 - <c:forEach> loop
2. Make two files as follows:
 - a. main.html: shows 2 text boxes and 3 radio buttons with values "addition", "subtraction" and "multiplication"

UG-Computer Science - CBCS Scheme

b. operate.jsp: depending on what the user selects perform the corresponding function (Give two implementations: using request.getParameter() and using expression language)

3. Validate User input entered in a form. The input must include Name, DOB, Email ID, Lucky Number, Favorite food etc. (Refer Chapter 8)

4. Display Good Morning <uname>, Good Afternoon <uname> or Good Evening <uname> based on the current time of the day.

5. Create your custom library which contains two tags: <hello>, <choco>.

Usage of the tags:

- <hello name="Ajay">: Output should be Hello Ajay. It contains a mandatory attribute 'name' which can accept Dynamic value.

- <choco texture="Chewy">: Output should be FiveStar, BarOne.

- <choco texture="Crunchy">: Output should be Munch. KitKat.

That means the mandatory attribute must accept a value, and

based on the attributes value, it should give output. You must use

a bean ChocoBean for this purpose.

SEMESTER VI

Course code: CMF25202

DSE6B: Elective:Computer Science - VI Mobile Applications

Credits: Theory - 04, Practical - 01

Theory: 60 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Deliberate the details of Concepts of Event Driven Programming
- CO2. Learn in details with examples issues of Mobile applications
- CO3. Specify the details of Mobile applications Development tools and Frameworks
- CO4. Deliberate in details with examples common Mobile device UI's
- CO5. Write down in depth Data persistence Remote data storage and communication

CO6. Learn in details with examples Code signing

Unit - 1

(15 Lectures)

Event Driven Programming: UI event loop, Threading for background tasks, Outlets / actions, delegation, notification, Model View Controller (MVC) design pattern.

Mobile application issues: limited resources (memory, display, network, file system), input / output (multi-touch and gestures), sensors (camera, compass, accelerometer, GPS)

Unit - 2

(15 Lectures)

Development tools: Apple iOS toolchain: Objective-C, Xcode IDE, Interface Builder, Device simulator.

Frameworks: Objective-C and Foundation Frameworks, Cocoa Touch, UIKit, Others: Core Graphics, Core Animation, Core Location and Maps, Basic Interaction.

UG-Computer Science - CBCS Scheme

Unit - 3

(15 Lectures)

Common UI's for mobile devices: Navigation Controllers, Tab Bars, Table Views, Modal views, UI Layout.

Data Persistence: Maintaining state between application invocations, File system, Property Lists, SQLite, Core Data.

Unit - 4

(15 Lectures)

Remote Data-Storage and Communication: "Back End" / server side of application, RESTful programming, HTTP get, post, put, delete, database design, server side JavaScript / JSON.

Code signing: security, Keychain, Developers and App Store License Agreement (6L)

Reference:

1. Rajiv Ramnath, Roger Crawfis, and Paolo Sivilotti, Android SDK 3 for Dummies, Wiley, 2011.
2. Valentino Lee, Heather Schneider, and Robbie Schell, Mobile Applications: Architecture, Design, and Development, Prentice Hall, 2004.
3. Brian Fling, Mobile Design and Development, O'Reilly Media, 2009. Maximiliano
4. Firtman, Programming the Mobile Web, O'Reilly Media, 2010.
5. Christian Crumlish and Erin Malone, Designing Social Interfaces, O'Reilly Media, 2009.
6. James E Smith, Ravi Nair, "Virtual Machines", Morgan Kaufmann Publishers, 2006.

Mobile Applications Lab

Software Lab based on Mobile Applications:

1. Installing Android Environment
2. Create Hello World Application
3. Sample Application about Android Resources
4. Sample Application about Layouts

UG-Computer Science - CBCS Scheme

5. Sample Application about Intents
6. Sample Application I about user interfaces
7. Sample Application about Animations
8. Make a Project based on above labs
9. Sample Application about Android Data
10. Sample Application about SQLite I
11. Sample Application about SQLite II
12. Project Presentation

SEMESTER VI

Course code: CMF25402

**DSE6C: Elective:Computer Science - VI
Cloud Computing**

Credits: Theory - 04, Practical - 01

Theory: 60 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Learn in depth Fundamentals of Cloud Computing
- CO2. Understand the details of Cloud Services and File System
- CO3. Learn in depth Concept of Collaborating with Cloud
- CO4. Understand the details of Virtualization in cloud
- CO5. Learn the classification and characteristics of Security challenges in Cloud Computing
- CO6. Specify the classification and characteristics of Security challenges in Cloud Computing
- CO7. Understand the details of Security challenges in Cloud Computing
- CO8. Understand the Common standards of Cloud Computing
- CO9. Deliberate in details with examples Various Application of Cloud Computing

Unit - 1

(15 Lectures)

Cloud Introduction: Cloud Computing Fundamentals: Cloud Computing definition, Types of cloud, Cloud services: Benefits and challenges of cloud computing, Evolution of Cloud Computing , usage scenarios and Applications , Business models around Cloud– Major Players in Cloud Computing - Issues in Cloud - Eucalyptus - Nimbus - Open Nebula, CloudSim.

Cloud Services And File System: Types of Cloud services: Software as a Service - Platform as a Service – Infrastructure as a Service - Database as a Service - Monitoring as a Service –Communication as services.

UG-Computer Science - CBCS Scheme

Unit - 2

(15 Lectures)

Service providers- Google App Engine, Amazon EC2, Microsoft Azure, Sales force.

Collaborating With Cloud:Collaborating on Calendars, Schedules and Task Management –Collaborating on Event Management, Contact Management, Project Management –Collaborating on Word Processing , Databases – Storing and Sharing Files- Collaborating viaWeb-Based Communication Tools – Evaluating Web Mail Services – Collaborating via SocialNetworks – Collaborating via Blogs and Wikis. 185 CS-Engg&Tech-SRM-2013

Unit - 3

(15 Lectures)

Virtualization For Cloud: Need for Virtualization – Pros and cons of Virtualization – Types of Virtualization – System Vm, Process VM, Virtual Machine monitor – Virtual machine properties - Interpretation and binary translation, HLL VM - Hypervisors – Xen, KVM , VMWare, Virtual Box, Hyper-V.

Unit - 4

(15 Lectures)

Security, Standards, And Applications: Security in Clouds: Cloud security challenges - Software as a Service Security, Common Standards: The Open Cloud Consortium - The Distributed management Task Force - Standardsfor application Developers - Standards for Messaging - Standards for Security, End user access to cloud computing, Mobile Internet devices and the cloud.

Reference:

1. Bloor R., Kanfman M., Halper F. Judith Hurwitz “Cloud Computing ” Wiley India Edition,2010
2. John Rittinghouse& James Ransome, “Cloud Computing Implementation Management and Strategy”, CRC Press, 2010
3. Antohy T Velte ,Cloud Computing : “A Practical Approach”, McGraw Hill,2009
4. Michael Miller, Cloud Computing: “Web-Based Applications That Change the Way You Work and Collaborate Online”, Que Publishing, August 2008.

UG-Computer Science - CBCS Scheme

5. James E Smith, Ravi Nair, "Virtual Machines", Morgan Kaufmann Publishers, 2006.

Online Reading/Supporting Material

1. Haley Beard, "Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing", Applications and Data Centers in the Cloud with SLAs, Emereo Pty Limited, July 2008
2. webpages.iust.ac.ir/hsalimi/.../89.../Cloud%20Common%20standards.ppt ennebula.org,
3. www.cloudbus.org/cloudsim/, <http://www.eucalyptus.com/>
4. hadoop.apache.org
5. http://hadoop.apache.org/docs/stable/hdfs_design.html
6. http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en//archive/mapreduce-osdi04.pdf

Cloud Computing Lab

Software Lab based on Cloud Computing:

1. Create virtual machines that access different programs on same platform.
2. Create virtual machines that access different programs on different platforms.
3. Exploring Google cloud for the following
 - a) Storage
 - b) Sharing of data
 - c) Manage your calendar, to-do lists,
 - d) A document editing tool
4. Exploring Open source cloud (Any two)

SEMESTER VI

Course code: CMF38002

SEC4A: Elective:Computer Science - VII

Android Programming

Credits: Theory - 01, Practical - 01

Theory: 15 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Learn the details of Overview of Android
- CO2. Understand in details with examples Concepts of OOP's Using JAVA
- CO3. Learn in depth Development Tools
- CO4. Understand in depth User Interface Architecture
- CO5. Learn the details of User Interface Design
- CO6. Understand in depth SQLite Database Connectivity

Unit - 1

(15 Lectures)

Introduction: History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture.

Overview of object oriented programming using Java: OOPs Concepts: Inheritance, Polymorphism, Interfaces, Abstract class, Threads, Overloading and Overriding, Java Virtual Machine.

Development Tools: Installing and using Eclipse with ADT plug-in, Installing Virtual machine for Android sandwich/Jelly bean (Emulator), configuring the installed tools, creating a android project – Hello Word, run on emulator, Deploy it on USB-connected Android device.

User Interface Architecture: Application context, intents, Activity life cycle, multiple screen sizes.

User Interface Design: Form widgets, Text Fields, Layouts, Button control, toggle buttons, Spinners (Combo boxes), Images, Menu, and Dialog.

Database: Understanding of SQLite database, connecting with the database.

UG-Computer Science - CBCS Scheme

Reference:

1. Android application development for java programmers. By James C. Sheusi. Publisher: Cengage Learning, 2013.

ONLINE READING / SUPPORTING MATERIAL:

1. <http://www.developer.android.com>
2. <http://developer.android.com/about/versions/index.html>
3. <http://developer.android.com/training/basics/firstapp/index.html>
4. <http://docs.oracle.com/javase/tutorial/index.htm> (Available in the form of free downloadable ebooks also).
5. <http://developer.android.com/guide/components/activities.html>
6. <http://developer.android.com/guide/components/fundamentals.html>
7. <http://developer.android.com/guide/components/intents-filters.html>.
8. <http://developer.android.com/training/multiscreen/screensizes.html>
9. <http://developer.android.com/guide/topics/ui/controls.html>
10. <http://developer.android.com/guide/topics/ui/declaring-layout.html>
11. <http://developer.android.com/training/basics/data-storage/databases.html>

Android Programming Lab

Software Lab Based on Android Programming:

1. Create “Hello World” application. That will display “Hello World” in the middle of the screen in the emulator. Also display “Hello World” in the middle of the screen in the Android Phone.
2. Create an application with login module. (Check username and password).
3. Create spinner with strings taken from resource folder (res >> value folder) and on changing the spinner value, Image will change.
4. Create a menu with 5 options and selected option should appear in text box.
5. Create a list of all courses in your college and on selecting a particular course teacher-in-charge of that course should appear at the bottom of the screen.
6. Create an application with three option buttons, on selecting a button colour of the screen will change.
7. Create and Login application as above. On successful login, pop up the message.
8. Create an application to Create, Insert, update, Delete and retrieve operation on the database.

SEMESTER VI

Course code: CMF38202

SEC4B: Elective:Computer Science - VII

PHP Programming

Credits: Theory - 01, Practical - 01

Theory: 15 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. Learn in depth Elements of PHP
- CO2. Learn in depth Interaction Methods Between HTML and PHP
- CO3. Understand in depth PHP function
- CO4. Understand in depth String Manipulation
- CO5. Learn the characteristics of Regular Expression
- CO6. Learn the details of Developing PHP Web Application

Unit - 1

(15 Lectures)

Introduction to PHP: PHP introduction, inventions and versions, important tools and software requirements (like Web Server, Database, Editors etc.), PHP with other technologies, scope of PHP, Basic Syntax, PHP variables and constants, Types of data in PHP , Expressions, scopes of a variable (local, global),, PHP Operators : Arithmetic, Assignment, Relational , Logical operators, Bitwise , ternary and MOD operator.

PHP operator Precedence and associativity

Handling HTML form with PHP: Capturing Form Data, GET and POST form methods Dealing with multi value fields, Redirecting a form after submission, PHP conditional events and Loops: PHP IF Else conditional statements (Nested IF and Else) Switch case, while ,For and Do While Loop, Goto , Break ,Continue and exit

PHP Functions: Function, Need of Function , declaration and calling of a function, PHP Function with arguments, Default Arguments in Function, Function argument with call by value, call by reference, Scope of Function Global and Local

UG-Computer Science - CBCS Scheme

String Manipulation and Regular Expression: Creating and accessing String , Searching & Replacing String, Formatting, joining and splitting String , String Related Library functions, Use and advantage of regular expression over inbuilt function, Use of preg_match(), preg_replace(), preg_split()

functions in regular expression Array: Anatomy of an Array ,Creating index based and Associative array ,Accessing array, Looping with Index based array, with associative array using each() and foreach(),Some useful Library function

Reference:

1. Rajiv Ramnath, Roger Crawfis, and Paolo Sivilotti, Android SDK 3 for Dummies, Wiley, 2011.
2. Valentino Lee, Heather Schneider, and Robbie Schell, Mobile Applications: Architecture, Design, and Development, Prentice Hall, 2004.
3. Brian Fling, Mobile Design and Development, O'Reilly Media, 2009. Maximiliano
4. Firtman, Programming the Mobile Web, O'Reilly Media, 2010.
5. Christian Crumlish and Erin Malone, Designing Social Interfaces, O'Reilly Media, 2009.

PHP Programming Lab

Software Lab Based on PHP:

1. Create a PHP page using functions for comparing three integers and print the largest number.
2. Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as an argument.
3. WAP to check whether the given number is prime or not.
4. Create a PHP page which accepts string from user. After submission that page displays the reverse of provided string.
5. Write a PHP function that checks if a string is all lower case.
6. Write a PHP script that checks whether a passed string is palindrome or not? (A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)

UG-Computer Science - CBCS Scheme

7. WAP to sort an array.
8. Write a PHP script that removes the whitespaces from a string.
Sample string : 'The quick " " brown fox'
Expected Output :Thequick""brownfox
9. Write a PHP script that finds out the sum of first n odd numbers.
10. Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.
11. Write a PHP script that checks if a string contains another string.
12. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birth day.
13. Create a script to construct the following pattern, using nested for loop.

```
*  
  
* *  
  
* * *  
  
* * * *  
  
* * * * *
```

14. Write a simple PHP program to check that emails are valid.
15. WAP to print first n even numbers.
16. \$color = array('white', 'green', 'red')

Write a PHP script which will display the colors in the following way :

Output :

white, green, red,

- green
- red
- white

UG-Computer Science - CBCS Scheme

17. Using switch case and dropdown list display a “Hello” message depending on the language selected in drop down list.
18. Write a PHP program to print Fibonacci series using recursion.
19. Write a PHP script to replace the first 'the' of the following string with 'That'.

Sample : 'the quick brown fox jumps over the lazy dog.'

Expected Result : That quick brown fox jumps over the lazy dog.

SEMESTER VI

Course code: CMF38402

**SEC4C: Elective:Computer Science - VII
System Administration and Maintenance**

Credits: Theory - 01, Practical - 01

Theory: 15 Lectures

Course Outcome:

After successful completion of the course, the student is able to

- CO1. System Administration & Maintenance
- CO2. Understand in details of Networking OS Environment
- CO3. Deliberate in depth Basic LINUX Commands
- CO4. Deliberate in details with examples Basic Windows Commands
- CO5. Learn the details of Server Configuration and Maintenance

Unit - 1

(15 Lectures)

Part I - Linux/Unix:Basics of operating system, services,Installation and configuration, maintenance. What is linux/unix Operating systems?, Kernel, API, cli, gui,Difference between linux/unix and other operating systems, Features and Architecture, Linux features, advantages, disadvantages

Part II - Windows:Windows as operating system, history, versions. PC hardware, BIOS, Devices and drivers, Kernal Configuration and building, Application installation, configuration and maintenance, Server services and Client services, Difference between WindowsXP/windows7 and windows server 2003/2008.

Reference:

1. Evi Nemeth, Garth Snyder and others,Unix and Linux System Administration Handbook, 4th Ed,Pearson Education, 2016.
2. PDaniel J. Barrett, Linux Pocket Guide: Essential Commands, 3rd Edition,Shroff/O'Reilly; Second edition 4 June 2012
3. Woody Leonhard,Windows 7 All-in-One For Dummies,2009

System Administration and Maintenance Lab

UG-Computer Science - CBCS Scheme

Linux:

Linux Desktop tour. Configuring desktop environment and desktop settings.

Basic Commands : Terminal, shell, Cat, ls, cd, date, cal, man, echo, pwd, Mkdir, rm, rmdir Ps, kill

Package Installation

Synaptic package manager

Windows:

Creating users – Admin and regular.

Path of their personal files. Adding and changing passwords.

Difference between workgroup and domain. Concept of roles.

user profiles – creating and roaming Concept of Active Directory. Creating active directory in windows 2003/2008.

Process and Disk management

Windows Task manager. File systems – NTFS, FAT.

Services

Control Panel

C:/program Files, C:/system C:/windows

Add /remove new hardware (like printer), Add/remove new programmes.

Network Administration

Ipconfig, Ping, tracert, route, hostname, net, netstat, whoami

Set manual IP address, check connectivity – ipv4, ipv6

Administrator Tools

Control Panel -> Administrative Tools

Computer Management, Local security Policy, Performance Monitor, Task Scheduler, Antivirus and firewall.

Misc

Start->Accessories->System tools ->All options (Remote desktop, backup/restore etc.)

LAN - sharing printer, files and folder over the network.

**Question Paper Pattern
Theory (4 Credits)**

Time: 3 Hours

Max. Marks: 70

Part – A

I. Answer any Eleven Question out of given Twelve Questions. 11 X 2 = 22

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
- 12.

Note: Three Questions each from the Units 1, 2, 3, and 4

Part - B

II. Answer any Two SubQuestions from each main Question.

- | | |
|--------|------------|
| 13. A) | 2 X 6 = 12 |
| B) | |
| C) | |
| 14. A) | 2 X 6 = 12 |
| B) | |
| C) | |
| 15. A) | 2 X 6 = 12 |
| B) | |
| C) | |
| 16. A) | 2 X 6 = 12 |
| B) | |
| C) | |

UG-Computer Science - CBCS Scheme

Note: Each Main questions contains 3 sub questions carries 06 Marks (may have internal Split-ups) and from Units 1, 2, 3 and 4 respectively

Question Paper Pattern

Theory (1 Credit)

Time: 2 Hours

Max. Marks: 50

Part – A

I. Answer all Questions.

05 X 02 = 10

- 1.
- 2.
- 3.
- 4.
- 5.

Part - B

II. Answer any Four Questions out of given Five Questions.

4 X 10 = 40

- 1.
- 2.
- 3.
- 4.
- 5.

Note: Each Main questions contains 3 sub questions carries 06 Marks (may have internal Split-ups) and from Units 1, 2, 3 and 4 respectively

UG-Computer Science - CBCS Scheme

Question Paper Pattern

Practical / Project (1 or 1.5 or 2 Credits)

Time: 3 Hours

Max. Marks: 70

Practicals

1. Two Experiments/ Programs 20 Marks Each 2 x 20 = 40 Marks
 - a. Write-ups 10 Marks
 - b. Conducting & Results 10 Marks (Any One for 2 Hours of Practical's)
2. Viva – Voice 10 Marks
3. Record 10 Marks

Project

1. Project Presentation 15 Marks
2. Project Demo 20 Marks
3. Viva – Voice 10 Marks
4. Record 25 Marks

Question Paper Pattern

Practical (1 Credit)

Time: 2 Hours

Max. Marks: 50

- a. Experiment/ Programs 30 Marks Each 30 Marks
- b. Conducting & Results 10 Marks
2. Viva – Voice 10 Marks
3. Record 10 Marks