

CHOICE BASED CREDIT SYSTEM

B.Sc. Program with Computer Science

Semester			Total credit
I	CORE	MSGCOR01T: Problem Solving with Computer MSGCOR01P: Problem Solving with Computer	4 2
	AECC1	ENVS	2
II	CORE	MSGCOR02T: DBMS MSGCOR02P: DBMS	4 2
	AECC2	English	2
III	CORE	MSGCOR03T: OS MSGCOR03P: LINUX	4 2
	SEC1	CMSSE01M: Programming in Python	6
IV	CORE	MSGCOR04T: Computer System Architecture MSGCOR04P: Computer System Architecture	4 2
	SEC2	CMSSE02M: R Programming	6
V	DSE1 (Any one)	MSGDSE01T: Programming in JAVA MSGDSE02T: Discrete Structures	6
VI	DSE2 (Any one)	MSGDSE03T: Software Engg MSGDSE04T: Computer Networks	6
Total number of courses		10	120

CORE PAPERS:

CMSGCOR01T : Problem Solving with Computer

Computer Fundamentals: Introduction to Computers: Characteristics of Computers, Uses of computers, Types and generations of Computers. (3L)

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

Planning the Computer Program: Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. (3L)

Techniques of Problem Solving: Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming. (4L)

Overview of Programming: Structure of a Python Program, Elements of Python (4L)

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). (8L)

Creating Python Programs: Input and Output Statements, Control statements (Looping- while Loop, for Loop , Loop Control, Conditional Statement- if...else, Difference between break, continue and pass). (10L)

Structures: Numbers, Strings, Lists, Tuples, Dictionary, Date & Time, Modules, Defining Functions, Exit function, default arguments. (10L)

Introduction to Advanced Python: Objects and Classes, Inheritance, Regular Expressions, Event Driven Programming, GUI Programming. (14L)

Reference Books:

1. P. K. Sinha & Priti Sinha , “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

6. <http://docs.python.org/3/tutorial/index.html>
7. <http://interactivepython.org/courselib/static/pythonds>
8. <http://www.ibiblio.org/g2swap/byteofpython/read/>

CMSGCOR01P: Problem Solving with Computer

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 :
 - a. Grade A: Percentage ≥ 80
 - b. Grade B: Percentage ≥ 70 and < 80
 - c. Grade C: Percentage ≥ 60 and < 70
 - d. Grade D: Percentage ≥ 40 and < 60
 - e. 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.
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:

$$P(t) = (15000(1+t))/(15 + e^t)$$
 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 ($s=(v*v-u*u)/2*a$)

CMSGCOR02T: Database Management Systems

Introduction to Database Management Systems: Characteristics of database approach, data models, DBMS architecture and data independence.

(10L)

Entity Relationship and Enhanced ER Modeling: Entity types, relationships, SQL-

99: Schema Definition, constraints, and object modeling. (15L)

Relational Data Model : Basic concepts, relational constraints, relational algebra, SQL queries.

(15L)

Database design: ER and EER to relational mapping, functional dependencies, normal forms up to third normal form. (20 L)

Books Recommended:

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.

CMSGCOR02P: Database Management Systems

Note: MyAccess/MySQL may be used.

The following concepts must be introduced to the students:

DDL Commands

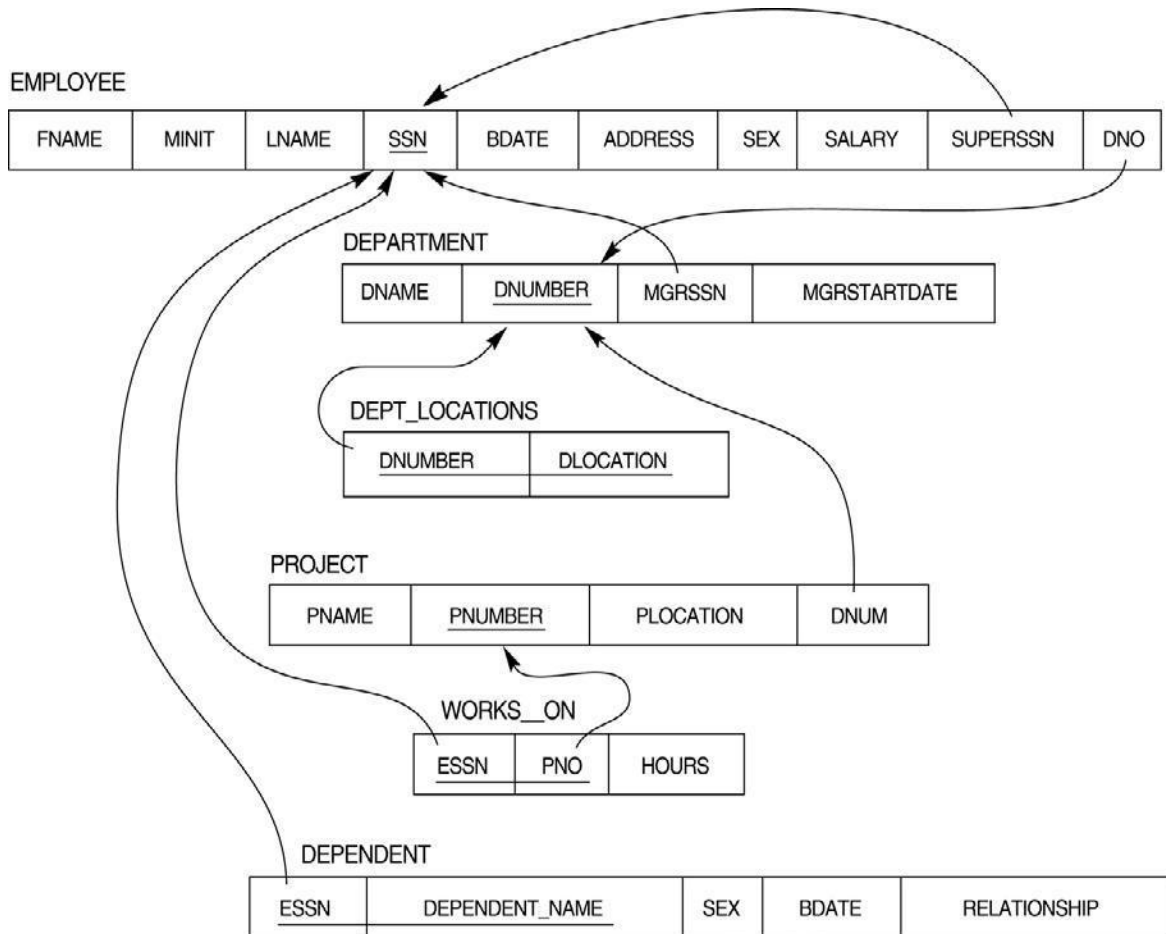
- Create table, alter table, drop table

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

Relational Database Schema - COMPANY



Questions to be performed on above schema

1. Create tables with relevant foreign key constraints
2. Populate the tables with data
3. Perform the following queries on the database :
 1. Display all the details of all employees working in the company.
 2. Display ssn, lname, fname, address of employees who work in department no 7.
 3. 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 10.
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.

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.

CMSGCOR03T: Operating Systems

Introduction: System Software, Resource Abstraction, OS strategies. (2L)

Types of operating systems - Multiprogramming, Batch, Time Sharing, Single user and

Multiuser, Process Control & Real Time Systems. (2L)

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. (10L)

Process Management : System view of the process and resources, initiating the OS, process address space, process abstraction, resource abstraction, process hierarchy, Thread model (15L) **Scheduling**: Scheduling Mechanisms, Strategy selection, non-pre-emptive and pre-emptive

strategies. (12L)

Memory Management: Mapping address space to memory space, memory allocation

strategies, fixed partition, variable partition, paging, virtual memory (12L)

Shell introduction and Shell Scripting

(7L)

- 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)

Books Recommended:

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.

MSGCOR03P: 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.
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 “*”.

CMSGCOR04T: Computer System Architecture

Introduction: Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexors, registers, counters and memory units. (12L)

Data Representation and basic Computer Arithmetic: Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison. (8L)

Basic Computer Organization and Design: Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt.(18L)

Central Processing Unit: Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control. (10L)

Programming the Basic Computer: Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming. (8L)

Input-output Organization: Peripheral devices, I/O interface, Modes of data transfer, direct memory access. (4L)

Books Recommended:

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

CMSGCOR04P: Computer System Architecture Lab
Periods

Practical: 60 Lab

Memory 4096 words 16 bits per word		0	Instruction format 3 4	15		
		Opcode	Address			

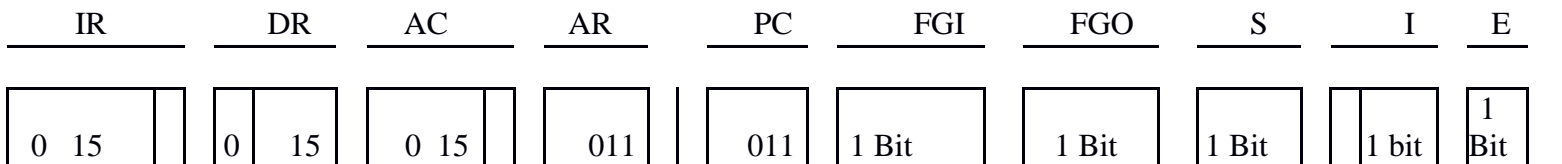
Basic Computer Instructions

Memory Reference

Register Reference

Input-Output

1. Create a machine based on the following architecture: Register Set



Symbol	Hex	Symbol	Hex	Symbol	Hex
					F80
AND	0xxx	CLA	E800	INP	0
					F40
ADD	2xxx	CLE	E400	OUT	0
ISZ	Cxxx	INC	E020		
AND_I	1xxx	SPA	E010		

ADD_I	3xxx	Indirect Addressing	SNA	E008				
LDA_I	5xxx		SZA	E004				
STA_I	7xxx		SZE	E002				
BUN_I	9xxx		HLT	E001				
BSA_I	Bxxx							
ISZ_I	Dxxx							

Refer to Chapter-5 of Morris Mano for description of instructions.

ii) Create the micro operations and associate with instructions as given in the chapter (except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.

iii) Create a Fetch routine of the instruction cycle.

iv) Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:

- a. CLA
- b. CLE
- c. CMA
- d. CME
- e. CIR
- f. CIL
- g. INC
- h. SPA
- i. SNA
- j. SZA
- k. SZE
- l. HLT

Initialize the contents of AC to (A937)₁₆, that of PC to (022)₁₆ and E to 1.

5. 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
- b. AND
- c. LDA
- d. STA
- e. BUN
- f. BSA
- g. ISZ

6. 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.

7. Modify the machine created in Practical 1 according to the following instruction format:

Instruction format

0	2	3	4	15
---	---	---	---	----



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

Discipline Specific Elective Papers:

CMSGDSE01T: Programming in Java

75 Lectures

Additional Tutorial: 15 Lectures

Introduction to Java: Features of Java, JDK Environment

(2L)

Object Oriented Programming Concept Overview of Programming, Paradigm, Classes, Abstraction, Encapsulation, Inheritance, Polymorphism, Difference between C++ and JAVA

(12L)

Java Programming Fundamental : Structure of java program, Data types, Variables, Operators, Keywords, Naming Convention, Decision Making (if, switch), Looping (for, while), Type Casting

(12L)

Classes and Objects: Creating Classes and objects, Memory allocation for objects, Constructor,

Implementation of Inheritance, Implementation of Polymorphism, Method Overloading, Method Overriding, Nested and Inner classes

(12L)

Arrays and Strings: Arrays, Creating an array, Types of Arrays, String class Methods, StringBuffer methods.

(8L)

Abstract Class, Interface and Packages: Modifiers and Access Control, Abstract classes and methods, Interfaces, Packages Concept, Creating user defined packages

(10L)

Exception Handling: Exception types, Using try catch and multiple catch, Nested try, throw, throws and finally, Creating User defined Exceptions.

(6L)

File Handling: Byte Stream, Character Stream, File IO Basics, File Operations, Creating file, Reading file, Writing File

(6L)

Applet Programming: Introduction, Types Applet, Applet Life cycle, Creating Applet, Applet tag

(7L)

Books Recommended:

1. Ivan Bayross, Web Enabled Commercial Application Development Using Html, Dhtml, javascript, Perl Cgi , BPB Publications, 2009.

2. Cay Horstmann, BIG Java, Wiley Publication , 3rd Edition., 2009
3. Herbert Schildt , Java 7, The Complete Reference, , 8th Edition, 2009.
4. E Balagurusamy , Programming with JAVA, TMH, 2007

CMSGDSE02T: Discrete Structures

75 Lectures

Additional Tutorial: 15 Lectures

Introduction: Introduction to Sets, Finite and Infinite Sets, Unaccountably Infinite Sets. Introduction to Functions and relations, Properties of Binary relations, Closure, Partial Ordering Relations. (15L)

Unit-II: Pigeonhole Principle, Permutation and Combinations, Mathematical Induction, Principle of Inclusion and Exclusion. (15L)

Unit-III : Asymptotic Notations (5)

Recurrence Relations: Introduction, Generating Functions, Linear Recurrence Relations with constant coefficients and their solution. (15L)

Graphs Theory: Basic Terminology of Graphs, Models and Types, Multigraphs, Weighted Graphs, Graph Representation. Graph Isomorphism Graph Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Basic Terminology of Trees, Properties of Trees, Spanning Trees. (15L)

Inference Theory: Introduction, Logical Connectives, Well Formed Formulas, Tautologies, Equivalence (10L)

Books Recommended:

1. C. L. Liu and D.P. Mohapatra, Elements of Discrete Mathematics, Third Edition, Tata McGraw Hill , 2008.
2. K. Rosen, Discrete Mathematics and Its Applications, Sixth Edition, Tata McGraw Hill, 2007.
3. T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, Third Edition, Prentice Hall of India, 2010.
4. J.P. Trembley, R. Manohar, Discrete Mathematical Structures with Application to Computer Science, First Edition, Tata McGraw Hill, 2001.
5. David Gries, Fred B. Schneider, A Logical Approach to Discrete Math, Springer, 2010

Online Reading/Supporting Material:

1. <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-fall-2005/>

CMSGDSE03T: Software Engineering
Additional Tutorial: 15 Lectures

75 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. (10L)

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. Analysis and modeling, data, functional and behavioral models, structured analysis and data dictionary. (12L)

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 software design, system design, real time executives, data acquisition system, monitoring and control system. (15L)

Software Configuration Management: The SCM process, Version control, Change control, Configuration audit, SCM standards. (10L)

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. (12L)

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. (10L)

Trends in Software Engineering: Reverse Engineering and Re-engineering – wrappers – Case Study of CASE tools. (6L)

Books Recommended:

1. Roger S. Pressman, Software engineering- A practitioner's Approach, McGraw-Hill
2. Ian Sommerville, Software engineering, Pearson education Asia, 6th edition, 2000.
3. Pankaj Jalote- An Integrated Approach to Software Engineering, Springer Verlag, 1997.

4. James F Peters and Witold Pedryez, "Software Engineering – An Engineering Approach", John Wiley and Sons, New Delhi, 2000.
5. Ali Behforooz and Frederick J Hudson, "Software Engineering Fundamentals", Oxford University Press, New Delhi, 1996.
6. Pfleeger, "Software Engineering", Pearson Education India, New Delhi, 1999.
7. Carlo Ghezzi, Mehdi Jazayari and Dino Mandrioli, "Fundamentals of Software Engineering", Prentice Hall of India, New Delhi, 1991.

CMSGDSE04:Computer Networks

75 Lectures

Additional Tutorial: 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. (20L)

Physical Layer : Cabling, Network Interface Card, Transmission Media Devices- Repeater, Hub, Bridge, Switch, Router, Gateway. (8L)

Data Link Layer : Framing techniques; Error Control; Flow Control Protocols; Shared media protocols - CSMA/CD and CSMA/CA. (10L)

Network Layer : Virtual Circuits and Datagram approach, IP addressing methods – Subnetting; Routing Algorithms (adaptive and non-adaptive) (10L)

Transport Layer: Transport services, Transport Layer protocol of TCP and UDP (8L)

Application Layer : Application layer protocols and services – Domain name system, HTTP, WWW, telnet, FTP, SMTP (12L)

Network Security : Common Terms, Firewalls, Virtual Private Networks (7L)

Books Recommended:

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.

Skill Enhancement Courses (Credit: 02 each)

CMSSEEC01M: Programming in Python

Planning the Computer Program: Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. (2L)

Techniques of Problem Solving: Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.

Overview of Programming : Structure of a Python Program, Elements of Python (2L) (3L)

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). (4L)

Creating Python Programs : Input and Output Statements, Control statements (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments. (4L)

Reference Books

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation www.python.org 2015
3. Allen Downey, Jeffrey Elkner, Chris Meyers, How to think like a computer scientist : learning with Python, Freely available online. 2012
4. <http://docs.python.org/3/tutorial/index.html>
5. <http://interactivepython.org/courselib/static/pythonds>
6. <http://www.ibiblio.org/g2swap/byteofpython/read/>

Software Lab Based on 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 users 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

1. Percentage < 40

- 1 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.
- 2 WAP to display the first n terms of Fibonacci series.
- 3 WAP to find factorial of the given number.
- 4 WAP to find sum of the following series for n terms: $1 - 2/2! + 3/3! - \dots - n/n!$
- 5 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.

4. Write a menu-driven program to create mathematical 3D objects I. curve
 4. sphere
- III. cone
- IV. arrow
 1. ring
- VI. cylinder.
 1. WAP to read n integers and display them as a histogram.
 2. WAP to display sine, cosine, polynomial and exponential curves.

 3. 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.
 4. 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 $t \geq 0$.
 5. 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.
 1. 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 ($s=u*t+0.5*a*t*t$)
 - a. distance wrt velocity ($s=(v*v-u*u)/2*a$)
 1. WAP to show a ball bouncing between 2 walls. (Optional)

CMSSEEC02M:R-Programming

Introduction: Overview and History of R, Getting Help, Data Types, Subsetting, Vectorized (5L)
Operations, Reading and Writing Data. (5L)

Control Structures, Functions, lapply, tapply, split, mapply, apply, Coding Standards. (5L)

Scoping Rules, Debugging Tools, Simulation, R Profiler. (5L)

ReferenceBook

1. W. N. Venables, D. M. Smith, An Introduction to R, R-core team, 2015

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