INSTITUTE OF ENGINEERING AND TECHNOLOGY
DEEN DAYAL UPADHYAYA GORAKHPUR UNIVERSITY, GORAKHPUR

COURSE STRUCTURE

Master of Computer and Applications

Based

on

AICTE MODEL CURRICULUM

[Effective from the Session: 2025-26]

Curriculum for Master of Computer Application (MCA)
Course Structure & Semester-wise Credit Distribution

Credit Distribution (Course Type) in MCA Program		
S.No.	**Course Type**	**Credit**
1	ESC	5
2	HMSC	6
3	PCC	50
4	PEC	12
6	Project work, Seminar and internship in industry or elsewhere	17
Total Credit		**90**

Abbreviation:
ESC: Engineering Science Course
HMSC: Humanities & Social Science including Management Courses
PCC: Professional Core Course
PEC: Professional Elective Course

Engineering Science Course (ESC)			
1.	MCA-101	Programming for Problem Solving	3+0
2.	MCA-105	Programming for Problem Solving Lab	0+2
Total			**5**
Humanities and Social Sciences including Management Courses (HSMC)			
1.	HSM101	Professional Communication	3+0
2.	HSM301	Organization Behavior	3+0
Total			**6**

Professional Core Course (PCC)			
S.No.	**Course Code**	**Course Title**	**Credit**
1.	MCA-102	Python Programming	3+0
2.	MCA-103	Operating Systems	3+0
3.	MCA-104	Discrete Mathematics	3+0
4.	MCA-105	Software Engineering	3+0
5.	MCA-152	Python Programming Lab	0+2
6.	MCA-201	Java Programming	3+0
7.	MCA-202	Data Structure	3+0
8.	MCA-203	Database Management System	3+0
9.	MCA-253	Database Management System Lab	0+2
10.	MCA-204	Computer Based Numerical and Statistical Techniques	3+0
11.	MCA-205	Theory of Computation	3+0
12.	MCA-206	Advanced Computer Architecture	3+0
13.	MCA-251	Java Programming Lab	0+2
14.	MCA-252	Data Structure Using C Lab	0+2
15.	MCA-301	Machine Learning	3+0
16.	MCA-302	Computer Network	3+0
17.	MCA-303	Design and Analysis of Algorithms	3+0
18.	MCA-304	Distributed System	3+0
Total			**50**

Work, Seminar and Internship/Industrial Training in Industry or Elsewhere			
S.No.	**Course Code**	**Course Title**	**Credit**
1.	MCAP-301	Mini -Project	**0+3**
2.	MCAI-301	Industrial Training	**0+1**
3.	MCAS-401	Seminar	**0+2**
4.	MCAP-401	Project II	**0+11**
Total			**17**

AEC and SEC Offered by the University for Implementation of NEP2020

(University Mandatory Course)

As per NEP2020, year wise credit requirements for the award of "Post Graduate diploma in Computer Science & Application", "Post Graduate Diploma in Computer Science application & Management"," Master of computer application" are given below:

First Year: After earning 46 credits in first year (22 credits in first semester and 24 credits in second semester), student will be eligible for **"Post Graduate diploma in Computer Science & Application"**.

Second Year (3rd Semester): After earning 68 credits (46 credits from first year and 22 credits in second year (22 credits in third semester), student will be eligible for the award of **"Post Graduate Diploma in Computer Science Application & Management"**.

Second Year (4th Semester): After earning 90 credits (46 credits from first year and 44 credits in second year), student will be eligible for the degree of **"Master of Computer Application"**.

Table1: Year wise credit requirement for award of Post Graduate Diploma in Computer Science & Application, Post Graduate Diploma in Computer Science application & Management", and Master of computer Application under the light of NEP2020.

After Year	Credit Requirement	Eligibility of
1st Year (1st Semester & 2nd Semester)	46	Post Graduate Diploma in Computer Science & Application
2nd Year (3rd Semester)	68	Post Graduate Diploma in Computer Science Application & Management.
2nd Year (4th Semester)	90	Master of Computer Application

Professional Elective Course (PEC)			
S.No.	**Semester**	**PEC-Elective**	**Credit**
1	III	Elective-I	3+0
2	IV	Elective-II	3+0
3	IV	Elective -II	3+0
4	IV	Elective -II	3+0
Total			**12**
Student has to select unique course(s) either from Elective-I or from Elective-II in the semester. Repetition of courses is not allowed.			

Professional Elective Courses (from SWAYAM) -1
Duration: 12 Weeks, **Credit:** 3

S.No.	Course Code	Course Id	Course Name	SME Name	Institute	Link
1.	MCAE-301	noc23-cs127	Cyber Security and Privacy	Prof. Saji K Mathew	IITM	https://onlinecourses.nptel.ac.in/noc23_cs127/ preview
2.	MCAE-302	noc23-cs77	Computer Vision	Prof. Jayanta Mukhopadhyay	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs77/p review
3.	MCAE-303	noc23-cs89	Cloud Computing	Prof. Soumya Kanti Ghosh	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs89/p review
4.	MCAE-304	noc23-cs114	C-Based VLSI Design	Prof. Chandan Karfa	IITG	https://onlinecourses.nptel.ac.in/noc23_cs114/ preview
5.	MCAE-305	noc23-cs92	Artificial Intelligence : Search Methods For Problem Solving	Prof.Deepak Khemani	IITM	https://onlinecourses.nptel.ac.in/noc23_cs92/p review
6.	MCAE-306	noc23-cs102	Parameterized Algorithms	Prof. Neeldhara Misra	noc23-cs102	https://onlinecourses.nptel.ac.in/noc23_cs102/ preview
7.	MCAE-307	noc23-cs107	Social Networks	Prof .Sudarshan Iyengar Prof. Yayati Gupta	IIT Ropar	https://onlinecourses.nptel.ac.in/noc23_cs107/ preview
8.	MCAE-308	noc23-cs111	Computational Complexity	Prof. Subrahmanyam Kalyanasundaram	IIT Hyderabad	https://onlinecourses.nptel.ac.in/noc23_cs111/ preview
9.	MCAE-309	noc23-cs68	Advanced Distributed Systems	Prof. Smruti Ranjan Sarangi	IITD	https://onlinecourses.nptel.ac.in/noc23_cs68/p review
10.	MCAE-310	noc23-cs84	Statistical Learning For Reliability Analysis	Prof.Monalisa Sarma	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs84/p review
11.	MCAE-311	noc23-cs75	Ethical Hacking	Prof.Indranil Sengupta	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs75/p review
12.	MCAE-312	noc23-cs91	Software Testing	Prof.Meenakshi D'Souza	IIITB	https://onlinecourses.nptel.ac.in/noc23_cs91/p review
13.	MCAE-313	noc23-cs126	Deep Learning for Compute r Vision	Prof. Vineeth N Balasubramanian	IITH	https://onlinecourses.nptel.ac.in/noc23_cs126/ preview
14.	MCAE-314	noc23-cs69	Privacy And Security In Online Social Media	Prof. Ponnurangam Kumaraguru	IIITH	https://onlinecourses.nptel.ac.in/noc23_cs69/p review

Note: In case of changes in the above Professional Elective Courses (from SWAYAM) -1, Dean /Director/ HoD will issue separate list of subjects.

Professional Elective Courses (from Department of CSE)-1

S.No.	Course Code	Course Name	Credit
1.	MCAE-301	Introduction to Software Testing	3+0
2.	MCAE-302	Wireless Network Systems	3+0
3.	MCAE-303	Artificial Intelligence	3+0
4.	MCAE-304	Data Mining and Data Warehousing	3+0
5.	MCAE-305	Advanced Database Management System	3+0
6.	MCAE-306	Design & Development of Mobile Applications	3+0
7.	MCAE-307	Design and Analysis of Parallel Algorithms	3+0
8.	MCAE-308	High Performance Computing	3+0

Note:

1. Students may take subjects from Professional Elective Courses (from Department of CSE)-1 or Professional Elective Courses (from SWAYAM) -1.
2. Repetition of courses is not allowed.
3. Execution of the courses offered by the department depends on the availability of faculty persons.

Professional Elective Courses (from SWAYAM) -2
Duration: 12 Weeks, **Credit:** 3

S.No.	Course Code	Course Id	Course Name	SME Name	Institute	Link
1.	MCAE-401	noc23-cs07	Advanced Computer Architecture	Prof. Smruti R.Sarangi	IITD	https://onlinecourses.nptel.ac.in/noc23_cs07
2.	MCAE-402	noc23-cs59	Hardware Security	Prof. Debdeep Mukhopadhyay	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs59
3.	MCAE-403	noc23-cs61	GPU Architectures And Programming	Prof. Soumyajit Dey	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs61
4.	MCAE-404	noc23-cs62	Foundations of Cyber Physical Systems	Prof. Soumyajit Dey	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs62
5.	MCAE-405	noc23-cs63	Selected Topics in Algorithms	Prof. Palash Dey	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs63
6.	MCAE-406	noc23-cs36	Affective Computing	Prof. Jainendra Shukla Prof. Abhinav Dhall	IIITD	https://onlinecourses.nptel.ac.in/noc23_cs36
7.	MCAE-407	noc23-cs51	Introduction To Internet Of Things	Prof. Sudip Misra	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs51
8.	MCAE-408	noc23-cs08	Data Analytics with Python	Prof. A. Ramesh	IITR	https://onlinecourses.nptel.ac.in/noc23_cs08
9.	MCAE-409	noc23-cs03	Foundations of Cryptography	Prof. Ashish Choudhury	IIIT Bangalore	https://onlinecourses.nptel.ac.in/noc23_cs03
10.	MCAE-410	noc23-cs23	Reinforcement Learning	Prof. Balaraman Ravindran	IITM	https://onlinecourses.nptel.ac.in/noc23_cs23
11.	MCAE-411	noc23-cs24	Deep Learning	Prof. Sudarshan Iyengar	IIT Ropar	https://onlinecourses.nptel.ac.in/noc23_cs24
12.	MCAE-412	noc23-cs45	Natural Language Processing	Prof. Pawan Goyal	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs45
13.	MCAE-413	noc23-cs47	Blockchain and its Applications	Prof. Sandip Chakraborty Prof. Shamik Sural	IITKGP	https://onlinecourses.nptel.ac.in/noc23_cs47

Note: In case of changes in the above subjects Professional Elective Courses (from SWAYAM) -2, Dean /Director/ HoD will issue separate list of subjects.

Professional Elective Courses (from Department of CSE)-2

S.No.	Course Code	Course Name	Credit
1.	MCAE-401	Compiler Design	3+0
2.	MCAE-402	Back-End Technologies	3+0
3.	MCAE-403	Front End Technologies	3+0
4.	MCAE-404	Mobile Computing	3+0
5.	MCAE-405	Introduction to Deep Learning	3+0
6.	MCAE-406	Computational Intelligence	3+0
7.	MCAE-407	Natural Language Processing	3+0
8.	MCAE-408	Parallel and Distributed Computing	3+0

Note:

4. Students may take subjects from Professional Elective Courses (from Department of CSE)-2 or Professional Elective Courses (from SWAYAM) -2.
5. Repetition of courses is not allowed.
6. Execution of the courses offered by the department depends on the availability of faculty persons.

Credit Distribution (Semester wise) (Minimum Common Programme)

S.No.	Semester Number	Credit
1	Semester I	22
2	Semester II	24
3	Semester III	22
4	Semester IV	22
Total Credit		**90**

Course Structure

Branch/Course: Computer Science & Engineering
Semester-I (First Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Engineering Science Course	MCA-101	Problem solving using C	3+0
2.	Professional Core Course	MCA-102	Python Programming	3+0
3.	Professional Core Course	MCA -103	Operating Systems	3+0
4	Professional Core Course	MCA-104	Discrete Mathematics	3+0
5.	Professional Core Course	MCA-105	Software Engineering	3+0
6.	Humanities and Social Sciences including Management Courses	HSM101	Professional Communication	3+0
7.	Engineering Science Course	MCA-151	Problem Solving using C lab	0+2
8.	Professional Core Course	MCA-152	Python Programming Lab	0+2
Total Credit				**22**

Semester-II (First Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Professional Core Course	MCA-201	Java Programming	3+0
2.	Professional Core Course	MCA-202	Data structure	3+0
3.	Professional Core Course	MCA-203	Database Management System	3+0
4.	Professional Core Course	MCA-204	Computer Based Numerical and Statistical Techniques	3+0
5.	Professional Core Course	MCA-205	Theory of Computation	3+0
6.	Professional Core Course	MCA-206	Advanced Computer Architecture	3+0
7.	Professional Core Course	MCA-251	Java Programming Lab	0+2
8.	Professional Core Course	MCA-252	Data Structure using C Lab	0+2
9.	Professional Core Course	MCA-253	Database Management System Lab	0+2
Total Credit				**24**

Branch/Course: Computer Science & Engineering

Branch/Course: Computer Science & Engineering
Semester-III (Second Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Professional Core Course	MCA-301	Machine Learning	3+0
2.	Professional Core Course	MCA-302	Computer Network	3+0
3.	Professional Core Course	MCA-303	Design and Analysis of Algorithm	3+0
4.	Professional Core Course	MCA-304	Distributed System	3+0
5.	Professional Elective Course	MCAE-***	Elective-1 (from SWAYAM) or Elective-1(from Department of CSE)	3+0
6.	Humanities and Social Sciences including Management Courses	HSM301	Organization Behavior	3+0
7.	Professional Core Course	MCAI-301	Industrial /Practical Training	0+1
8.	Professional Core Course	MCAP-301	Mini-Project	0+3
Total Credit				**22**

Semester-IV (Second Year) Curriculum

S. No.	Type of Course	Course Code	Course Title	Credit
1.	Professional Elective Course	MCAE-***	Elective-2 (from SWAYAM) or Elective-2 (from Department of CSE)	3+0
2.	Professional Elective Course	MCAE-***	Elective-2 (from SWAYAM) or Elective-2 (from Department of CSE)	3+0
3.	Professional Elective Course	MCAE-***	Elective-2 (from SWAYAM) or Elective-2 (from Department of CSE)	3+0
4.	Professional Core Course	MCAS-401	Seminar	2+0
5.	Professional Core Course	MCAP-401	Project	11
Total Credit				**22**

Problem Solving using C (MCA-101)

Credit: 3+0

Course Objective: Students will be able to develop logics which will help them to create programs, applications in C also by learning the basic programming constructs they can easily switch over to any other language in future.

Syllabus

1. **Introduction to Programming:** Introduction to components of a computer system, Algorithm: Representation of Algorithm, Flowchart, Pseudo code with examples, from algorithms to programs, source code. Programming Basics: Structure of C program: writing and executing the first C program, Syntax and logical errors in compilation, object and executable code. Components of C language: Standard I/O in C, Fundamental data types, Variables and memory locations, Storage classes. Arithmetic expressions & Conditional Branching: Arithmetic expressions and precedence: Operators and expression using numeric and relational operators, mixed operands, type conversion, logical operators, bit operations, assignment operator, operator precedence and associatively. Conditional Branching: Applying if and switch statements, nesting if and else, use of break and default with switch case.

2. **Loops & Functions:** Iteration and loops: use of while, do while and for loops, multiple loop variables, use of break and continue statements. Functions: Introduction, types of functions, functions with array, passing parameters to functions, call by value, call by reference, recursive.

3. **Arrays & Basic Algorithms:** Array notation and representation, manipulating array elements, using multi-dimensional arrays. Character arrays and strings, Structure, union, enumerated data types, Array of structures, passing arrays to functions. Basic Algorithms: Searching &Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, Notion of order of complexity.

4. **Pointer& File Handling:** Pointers: Introduction, declaration, applications, Introduction to dynamic memory allocation (malloc, calloc, realloc, free), Use of pointers in self-referential structures, notion of linked list (no implementation) File handling: File I/O functions, Standard C pre-processors, defining and calling macros, command-line arguments.

Course Outcomes (CO): At the end of this course students will be able to:

CO1: To develop simple algorithms for arithmetic and logical problems.

CO2: To translate the algorithms to programs & execution (in C language).

CO3: To implement conditional branching, iteration and recursion.

CO4: To decompose a problem into functions and synthesize a complete program using divide and conquer approach.

CO5: To use arrays, pointers and structures to develop algorithms and programs.

Text Books:

1. The C programming by Kernighan Brain W. and Ritchie Dennis M., Pearson Education.
2. Computer Basics and C Programming by V.Rajaraman , PHI Learning Pvt. Limited, 2015.
3. Computer Concepts and Programming in C, E Balaguruswami, McGraw Hill
4. Computer Science- A Structured Programming Approach Using C, by Behrouz A. Forouzan, Richard F. Gilberg, Thomson, Third Edition , Cengage Learning - 2007.
5. Let Us C By Yashwant P. Kanetkar,BPB Publication.
6. Problem Solving and Program Design in C, by Jeri R. Hanly, Elliot B. Koffman, Pearson Addison-Wesley, 2006.
7. Programming in C by Kochan Stephen G. Pearson Education – 2015.
8. Computer Concepts and Programming by Anami, Angadi and Manvi, PHI Publication.

Problem Solving using C Lab (MCA-151)

Credit: 0+2

Note: A minimum of twenty experiments from the following should be performed.

1. Write a program to calculate the area of triangle using formula at=√s(s- a)(s-b)(s-c)
2. Basic salary of an employee is input through the keyboard. The DA is 25% of the basic salary while the HRA is 15% of the basic salary. Provident Fund is deducted at the rate of 10% of the gross salary (BS+DA+HRA).
3. Write a program to determine the roots of quadratic equation.
4. Write a program to find the largest of three numbers using nested if else.
5. Write a program to receive marks of physics, chemistry & maths from user & check its eligibility for course if
   - ***a.*** Marks of physics > 40
   - ***b.*** Marks of chemistry > 50
   - ***c.*** Marks of math's > 60
   - ***d.*** Total of physics & math's marks > 150 or
   - ***e.*** Total of three subjects marks > 200
6. Write a program to find the value of y for a particular value of n. The a, x, b, n is input by user
   - ***a.*** if n=1 y=ax%b if n=2 y=ax2+b2 if n=3 y=a-bx
   - ***b.*** if n=4 y=a+x/b
7. Write a program to construct a Fibonacci series up to n term.
8. Write a program to find whether the number is Armstrong number.
9. Write a program to generate sum of series 1!+2!+3!+ n!
10. Write a program to find the sum of following series 1-X1/1!+X2/2!……Xn/n!.
11. Write a program to print the entire prime no between 1 and 300.
12. Write a program to print out all the Armstrong number between 100 and 500.
13. Write a program to draw the following figure:

```
3 2 1
2 1
1


*
**
***
```

14. Write a program to receive a five-digit no and display as like 24689:
    - ***a.*** 2
    - ***b.*** 4
    - ***c.*** 6
    - ***d.*** 8
    - ***e.*** 9
15. Write a function that return sum of all the odd digits of a given positive no entered through keyboard.
16. Write a program to print area of rectangle using function & return its value to main function.
17. Write a program to calculate the factorial for given number using function.
18. Write a program to find sum of Fibonacci series using function.
19. Write factorial function & use the function to find the sum of series S=1!+2!+ n!.
20. Write a program to find the factorial of given number using recursion.

21. Write a program to find the sum of digits of a 5 digit number using recursion.
22. Write a program to calculate the GCD of given numbers using recursion.
23. Write a program to convert decimal number in to binary number.
24. Write a program to convert binary number into decimal number.
25. Write a program to delete duplicate element in a list of 10 elements & display it on screen.
26. Write a program to merge two sorted array & no element is repeated during merging.
27. Write a program to evaluate the addition of diagonal elements of two square matrixes.
28. Write a program to find the transpose of a given matrix & check whether it is symmetric or not.
29. Write a program to print the multiplication of two N*N (Square) matrix.
30. Write a program in C to check whether the given string is a palindrome or not.
31. Write program to sort the array of character (String) in alphabetical order like STRING in GINRST.
32. Write a program to remove all the blank space from the string & print it, also count the no of characters.
33. Write a program to store the following string “zero”, “one” “five”. Print the no in words, given in figure as 3205.
34. Write a program to compare two given dates. To store a date uses a structure that contains three members namely day, month and year. If the dates are equal, then display message equal otherwise unequal.
35. Define a structure that can describe a hotel. It should have the member that includes the name, address, grade, room charge and number of rooms.Write a function to print out hotel of given grade in order of room charges.
36. Define a structure called cricket with player name, team name, batting average, for 50 players & 5 teams. Print team wise list contains names of player with their batting average.
37. Write a c program to copy & count the character content of one file says a.txt to another file b.txt.
38. Write a program to take 10 integers from file and write square of these integer in another file.
39. Write a program to read number from file and then write all ‘odd’ number to file ODD.txt & all even to file EVEN.txt.
40. Write a program to print all the prime number, between 1 to 100 in file prime.txt.
41. Write the following C program using pointer:
    - ***a.*** To sort the list of numbers through pointer
    - ***b.*** To reverse the string through pointer.
42. Write a program to find the largest no among 20 integers array using dynamic memory allocation.
43. Using Dynamic Memory Allocation, write a program to find the transpose of given matrix.
44. Write a program to find the factorial of given number using command line argument.
45. Write a program to find the sum of digits of a 5-digit number using command line argument.
46. Write a program to implement Singly Linked List.
47. Write a program to implement Stack using dynamic memory allocation.
48. Write a program to implement Queue using dynamic memory allocation.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Python Programming (MCA-102)

Credit: 3+0

Course Objectives: This course enables students to solve problems in Python.

Syllabus

1. **Programming Basics and Decision Making :** Key features and applications of Python, Python Editors and Compilers (Interpreters), Using different offline and online Python IDE, Interacting with Python programs, **Data types:** Numeric, Boolean, Strings, Lists, Sets, Tuples, Dictionary; **Variables:** Declaration and initialization; **Other concepts:** Operators, Expressions, Indentation, Comments, Casting, Simple Statements: Taking inputs from user, Displaying outputs; Conditional statements: If...Else. **Introduction to Linux:** Understanding Linux**,** Installation of Python on Linux

2. **Control Flow and Other Programming Concepts** For Loops, While Loops, Break, Continue; **Array:** Looping Array elements, Array methods; Functions: Local and Global Variables, Built-in functions, User defined functions, Declaration of a function, Defining the function, Calling of the function, Functions with arguments, Recursion.

3. **OOP and File Handling:** Classes and objects, attributes and methods, constructors and destructors, inheritance, polymorphism, **Exception Handling:** Try...Except; Management of text files: Type of files, various file operations on text files, creating a text file, opening a file, closing a file, reading a text file, writing into a text file, copying a file to another file.

4. **Advance Concepts:** Problem solving- Use of Python to solve real time problems, How Python helps to research problems, Creating various types of graphs corresponding to any data to show different kinds of results and analysis; **Data Analysis:** Understanding problems of data science and machine learning, Creating codes in Python for various data analysis problems, Other advance programs.

Course Outcomes:

CO1: Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.

CO2: Demonstrate proficiency in handling Strings and File Systems.

CO3: Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.

CO4: Interpret the concepts of Object-Oriented Programming as used in Python.

CO5: Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

Text Books:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.

2. Think Python, Allen Downey, Green Tea Press
3. Introduction to Python, Kenneth A. Lambert, Cengage
4. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
5. Learning Python, Mark Lutz, O'Really
6. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist '', 2nd edition, Updated for Python 3, Shroff/O 'Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/)
7. Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
8. John V Guttag, —Introduction to Computation and Programming Using Python '', Revised and expanded Edition, MIT Press, 2013
9. Robert Sedgewick, Kevin Wayne, Robert Dondero, —Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.
10. Timothy A. Budd, —Exploring Python, Mc-Graw Hill Education (India) Private Ltd., 2015.
11. Kenneth A. Lambert, —Fundamentals of Python: First Programs, CENGAGE Learning, 2012.
12. Charles Dierbach, —Introduction to Computer Science using Python: A Computational ProblemSolving Focus, Wiley India Edition, 2013.
13. Paul Gries, Jennifer Campbell and Jason Montojo, —Practical Programming: An Introduction to Computer Science using Python 3, Second edition, Pragmatic Programmers, LLC, 2013.
14. Christopher Negus, LINUX BIBLE , 10th Edition, Wiley.

Python Programming Lab (MCA-152)

Credit: 0+2

1. Krishna and his five friends have decided to go for an industrial visit by sharing the expenses of the fuel equally. Write a Python program to calculate the amount (in Rs) each of them needs to put in for the complete journey. The program should also display **True**, if the amount to be paid by each person is divisible by 3, otherwise it should display **False**.
   Hint: Use the relational operators in print statement.
   Assumptions: Assume that mileage of the vehicle, amount per litre of fuel and distance for one way are given.

2. Write a python program to find the best of two test average marks out of three test's marks accepted from the user.

3. Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number.

4. Defined as a function F as $F_n = F_{n-1} + F_{n-2}$. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed.

5. Develop a Python program to convert binary to decimal, octal to hexadecimal using functions.

6. Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.

7. Write a Python program to implement insertion sort and merge sort using lists.

8. Write a program to convert roman numbers in to integer values using dictionaries.

9. Write a function called isphonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression.

10. Develop a Python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com).

11. Write a Python program to accept a file name from the user and perform the following operations
    - ***a.*** Display the first N line of the file.
    - ***b.*** Find the frequency of occurrence of the word accepted from the user in the file.

12. Write a Python program to create a ZIP file of a particular folder which contains several files inside it.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Operating Systems (MCA-103)

Credit: 3+0

Course Objective: Student will learn about processes and processor management, concurrency and synchronization, memory management schemes, and secondary storage management, security and protection.

Syllabus

1. **Introduction:** Operating system and functions, Classification of Operating systems- Batch, Interactive, Time-sharing, Real-Time System, Multiprocessor Systems, Multiuser Systems, Multiprocess Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Reentrant Kernels, Monolithic and Microkernel Systems.

2. **Concurrent Processes:** Process Concept, Principle of Concurrency, Producer / Consumer Problem, Mutual Exclusion, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Test and Set operation; Classical Problem in Concurrency- Dining Philosopher Problem, Sleeping Barber Problem; Inter Process Communication models and Schemes, Process generation.

3. **CPU Scheduling & Disk Scheduling:** Scheduling Concepts, Performance Criteria, Process States, Process Transition Diagram, Schedulers, Process Control Block (PCB), Process address space, Process identification information, Threads and their management, Scheduling Algorithms, Multiprocessor Scheduling. Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock. Disk Scheduling: I/O devices, and I/O subsystems, I/O buffering, Disk storage and disk scheduling.

4. **Memory Management:** Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Understand the structure and functions of OS.

CO2: Learn about Processes, Threads and Scheduling algorithms.

CO3: Understand the principles of concurrency and Deadlocks.

CO4: Learn various memory management schemes.

CO5: Study I/O management and File systems.

Text Books:

1. Silberschatz, Galvin and Gagne, "Operating Systems Concepts", Wiley.
2. Sibsankar Halder and Alex A Aravind, "Operating Systems", Pearson Education.
3. Harvey M Dietel, "An Introduction to Operating System", Pearson Education.
4. D M Dhamdhere, "Operating Systems: A Concept based Approach", 2nd Edition, TMH.
5. William Stallings, "Operating Systems: Internals and Design Principles", 6th Edition, Pearson .Education

Discrete Mathematics (MCA-104)

Credit: 3+0

Course Objective: The objective of this course is to teach students how to think logically and mathematically. The course stresses on mathematical reasoning and describes different ways in which mathematical problems could be solved. There are four thematic areas covered in this course: mathematical reasoning, combinatorial analysis, discrete structures, and mathematical modelling.

Syllabus

1. **Set Theory:** Introduction, Combination of sets, Multisets, Ordered pairs. Proofs of some general identities on sets. Relations: Definition, Operations on relations, Properties of relations, Composite Relations, Equality of relations, Recursive definition of relation, Order of relations. Functions: Definition, Classification of functions, Operations on functions, recursively defined functions. Growth of Functions. Natural Numbers: Introduction, Mathematical Induction, Variants of Induction, Induction with Nonzero Base cases. Proof Methods, Proof by counter – example, Proof by contradiction.

2. **Algebraic Structures & Lattices:** Definition, Groups, Subgroups and order, Cyclic Groups, Cosets, Lagrange's theorem, Normal Subgroups, Permutation and Symmetric groups, Group Homomorphisms, Definition and elementary properties of Rings and Fields. Lattices: Definition, Properties of lattices – Bounded, Complemented, Modular and Complete lattice. Boolean Algebra: Introduction, Axioms and Theorems of Boolean algebra, Algebraic manipulation of Boolean expressions. Simplification of Boolean Functions, Karnaugh maps, Logic gates, Digital circuits and Boolean algebra.

3. **Propositional Logic:** Proposition, well-formed formula, Truth tables, Tautology, Satisfiability, Contradiction, Algebra of proposition, Theory of Inference. Predicate Logic: First order predicate, well-formed formula of predicate, quantifiers, Inference theory of predicate logic.

4. **Trees:** Definition, Binary tree, Binary tree traversal, binary search tree. Graphs: Definition and terminology, Representation of graphs, Multigraphs, Bipartite graphs, planar graphs, Isomorphism and Homeomorphism of graphs, Euler and Hamiltonian paths, Graph coloring, Recurrence Relation & Generating function: Recursive definition of functions, Recursive algorithms, Method of solving recurrences. Combinatorics: Introduction, Counting Techniques, Pigeonhole Principle

Course Outcome (CO): At the end of course, the student will be able to

CO1: Write an argument using logical notation and determine if the argument is or is not valid.

CO2: Understand the basic principles of sets and operations in sets.

CO3: Demonstrate an understanding of relations and functions and be able to determine their properties.

CO4: Demonstrate different traversal methods for trees and graphs.

CO5: Model problems in Computer Science using graphs and trees.

Text Books:

1. Koshy, Discrete Structures, Elsevier Pub. 2008 Kenneth H. Rosen, Discrete Mathematics and Its Applications, 6/e, McGraw-Hill, 2006.
2. B. Kolman, R.C. Busby, and S.C. Ross, Discrete Mathematical Structures, 5/e, Prentice Hall, 2004.
3. E. Scheinerman, Mathematics: A Discrete Introduction, Brooks/Cole, 2000.
4. R.P. Grimaldi, Discrete and Combinatorial Mathematics, 5/e, Addison Wesley, 2004
5. Liptschutz, Seymour, "Discrete Mathematics", McGraw Hill.
6. Trembley, J.P & R. Manohar, "Discrete Mathematical Structure with Application to Computer Science", McGraw Hill.
7. Deo, Narsingh, "Graph Theory With application to Engineering and Computer.Science.", PHI.
8. Krishnamurthy, V., "Combinatorics Theory & Application", East-West Press Pvt. Ltd., New Delhi

Software Engineering (MCA-105)

Credit: 3+0

Course Objective:

- To provide the idea of decomposing the given problem into Analysis, Design, Implementation, Testing and Maintenance phases.
- To provide an idea of using various process models in the software industry according to given circumstances.
- To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project

Syllabus

1. **Introduction:** Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Software Quality Attributes. Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models. Software Requirement Specifications (SRS): Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modelling, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS. Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.

2. **Software Design:** Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.

3. **Software Testing:** Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards.

4. **Software Maintenance and Software Project Management:** Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re- Engineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Estimation of Various Parameters such as Cost, Efforts,

Course Outcome (CO): At the end of course, the student will be able to

CO1: Explain various software characteristics and analyse different software Development Models.

CO2: Demonstrate the contents of a SRS and apply basic software quality assurance practices to ensure that design, development meet or exceed applicable standards.

CO3: Compare and contrast various methods for software design

CO4: Formulate testing strategy for software systems, employ techniques such as unit testing, Test driven development and functional testing.

CO5: Manage software development process independently as well as in teams and make use of Various software management tools for development, maintenance and analysis.

Text Books:

1. RS Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. Pankaj Jalote, Software Engineering, Wiley
3. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
4. KK Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.
5. Ghezzi, M. Jarayeri, D. Manodrioli, Fundamentals of Software Engineering, PHI Publication.
6. Ian Sommerville, Software Engineering, Addison Wesley.
7. Kassem Saleh, "Software Engineering", Cengage Learning.
8. P fleeger, Software Engineering, Macmillan Publication

Java Programming (MCA-201)

Credit: 3+0

Objectives:

This subject aims to introduce students to the Java programming language. Upon successful completion of this subject, students should be able to create Java programs that leverage the object-oriented features of the Java language, such as encapsulation, inheritance and polymorphism; use data types, arrays and other data collections; implement error-handling techniques using exception handling, create and event-driven GUI using Swing components.

UNIT-I OOP Concepts:-Data abstraction, encapsulation, inheritance, Polymorphism, classes and objects, Procedural and object oriented programming paradigms. Java Programming- History of Java, comments, Data types, Variables, Constants, Scope and Lifetime of variables, Operators, Type conversion and casting, Enumerated types, Control flow block scope, conditional statements, loops, break and continue statements, arrays, simple java stand alone programs, class, object, and its methods constructors, methods, static fields and methods, access control, this reference, overloading constructors, recursion, exploring string class, garbage collection

UNIT – II Inheritance- Inheritance types, super keyword, preventing inheritance: final classes & methods. Polymorphism- Method overloading and method overriding, abstract classes and methods. Interfaces- Interfaces Vs Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface, inner classes Packages- Defining, creating and accessing a package, importing packages.

UNIT-III Exception handling-Benefits of exception handling, the classification of exceptions exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, creating own exception subclasses. Multithreading – Differences between multiple processes and multiple threads, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter-thread communication, producer consumer problem.

UNIT-IV Collection Framework in Java – Introduction to java collections, Overview of java collection framework, Generics, Commonly used collection classes- Array List, Vector, Hash table, Stack, Lambda Expressions. Files- Streams- Byte streams, Character streams, Text input/output, Binary input/output, random access file operations, File management using File class. Connecting to Database – JDBC Type 1 to 4 drivers, Connecting to a database, querying a database and processing the results, updating data with JDBC, Data Access Object (DAO)

Course Outcomes: After completion of this course, student will be able to

CO1: Understand the fundamental concepts of Object-Oriented Programming (OOP), including data abstraction, encapsulation, inheritance, polymorphism, and the differences between procedural and object-oriented programming paradigms.

CO2: Develop Java programs using core Java concepts such as variables, operators, control flow statements, arrays, classes, objects, constructors, methods, static members, and string manipulation while managing memory through garbage collection.

CO3: Implement object-oriented features such as inheritance, polymorphism, method overloading, method overriding, abstract classes, interfaces, inner classes, and packages to design modular and reusable Java applications.

CO4: Apply exception handling techniques to develop robust Java programs, manage multithreading operations for concurrent execution, and implement thread synchronization for inter-thread communication.

CO5: Utilize Java's Collection Framework, generics, and commonly used collection classes while handling file operations using Java I/O streams, and establish database connectivity using JDBC to interact with relational databases efficiently.

TEXT BOOK:

1. Java Fundamentals – A Comprehensive Introduction, Herbert Schildt and Dale Skrien, TMH.

REFERENCE BOOKS:

1. Java for Programmers, P.J.Deitel and H.M.Deitel, PEA (or) Java: How to Program , P.J.Deitel and H.M.Deitel,PHI

2. Object Oriented Programming through Java, P. Radha Krishna, Universities Press.

3. Thinking in Java, Bruce Eckel,PE

4. Programming in Java, S. Malhotra and S. Choudhary, Oxford Universities Press.

Java Programming Lab (MCA-251)

Credit: 0+2

1. Write a java program to find the Fibonacci series using recursive and non-recursive functions
2. Write a java program to multiply two given matrices.
3. Write a java program that reads a line of integers and displays each integers and the sum of all integers use String Tokenizer
4. Write a java program that checks whether a given string is palindrome or not

5. Write an applet program that displays a simple message
6. Write a Java program compute factorial value using Applet

7. Write a java program that works as a simple calculator. Use a Grid Layout to arrange Buttons for digits and for the + - * % operations. Add a text field to display the result.
8. Write a Java program for display the exception in a message dialog box
9. Write a Java program that implements a multi-thread application that has three threads

10. Write a java program that connects to a database using JDBC
11. Write a java program to connect to a database using JDBC and insert values into it
12. Write a java program to connect to a database using JDBC and delete values from it
13. Write a java program to simulate a traffic light

14. Write a java program to create an abstract class named shape that contains an empty method named number of sides (). Provide three classes named trapezoid, triangle and Hexagon such that each one of the classes extends the class shape. Each one of the class contains only the method number of sides ()
15. that shows the number of sides in the given geometrical figures.

16. Write a java program to display the table using labels in Grid layout
17. Write a java program for handling mouse events
18. Write a Java program loads phone no, name from a text file using hash table
19. Implement the above program to load phone no, name from database insteadof text file

DATA STRUCTURE (MCA-202)

Credit: 3+0

Course Objective: To provide the knowledge of basic data structures and its implementations in context of writing efficient programs.

Syllabus

1. **Introduction:** Basic Terminology, Elementary Data Organization, Built in Data Types in C. Algorithm, Efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations: Big Oh, Big Theta and Big Omega, Time-Space trade-off. Abstract Data Types (ADT) Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Derivation of Index Formulae for 1-D,2-D,3-D and n-D Array Application of arrays, Sparse Matrices and their representations. Linked lists: Array Implementation and Pointer Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition Subtraction & Multiplications of Single variable & two variables Polynomial.

2. **Stacks and Queues:** Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack in C, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Iteration and Recursion- Principles of recursion, Tail recursion, Removal of recursion Problem solving using iteration and recursion with examples such as binary search, Fibonacci numbers, and Hanoi towers. Tradeoffs between iteration and recursion. Queues: Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue.

3. **Searching & Sorting:** Concept of Searching, Sequential search, Index Sequential Search, Binary Search. Concept of Hashing & Collision resolution Techniques used in Hashing. Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Merge Sort, Heap Sort and Radix Sort.

4. **Trees & Graphs:** Basic terminology used with Tree, Binary Trees, Binary Tree Representation: Array Representation and Pointer (Linked List) Representation, Binary Search Tree, Strictly Binary Tree, Complete Binary Tree. Extended Binary Trees, Tree Traversal algorithms: Inorder, Preorder and Postorder, Constructing Binary Tree from given Tree Traversal, Operation of Insertion, Deletion, Searching & Modification of data in Binary Search. Threaded Binary trees, Traversing Threaded Binary trees. Huffman coding using Binary Tree. Concept & Basic Operations for AVL Tree, B Tree & Binary Heaps. Graphs: Terminology used with Graph, Data Structure for Graph Representations: Adjacency Matrices, Adjacency List, Adjacency. Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Describe how arrays, linked lists, stacks, queues, trees, and graphs are represented in memory, used by the algorithms and their common applications.

CO2: Discuss the computational efficiency of the sorting and searching algorithms.

CO3: Implementation of Trees and Graphs and perform various operations on these data structure.

CO4: Understanding the concept of recursion, application of recursion and its implementation and removal of recursion.

CO5: Identify the alternative implementations of data structures with respect to its performance to solve a real-world problem.

Text Books:

1. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C and C++", PHI Learning Private Limited, Delhi India.
2. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publications Pvt Ltd Delhi India.
3. Lipschutz, "Data Structures" Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd.
4. Thareja, "Data Structure Using C" Oxford Higher Education.
5. AK Sharma, "Data Structure Using C", Pearson Education India.
6. Rajesh K. Shukla, "Data Structure Using C and C++" Wiley Dreamtech Publication.
7. Michael T. Goodrich, Roberto Tamassia, David M. Mount "Data Structures and Algorithms in C++",
   Wiley India.
8. P. S. Deshpandey, "C and Data structure", Wiley Dreamtech Publication.
9. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education.
10. Berztiss, AT: Data structures, Theory and Practice, Academic Press.
11. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with applications",
    McGraw Hill.
12. Adam Drozdek "Data Structures and Algorithm in Java", Cengage Learning.

Data Structure using C Lab (MCA-252)

Credit: 0+2

1. Write C Programs to illustrate the concept of the following:
2. Sorting Algorithms-Non-Recursive.
3. Sorting Algorithms-Recursive.
4. Searching Algorithm.
5. Implementation of Stack using Array.
6. Implementation of Queue using Array.
7. Implementation of Circular Queue using Array.
8. Implementation of Stack using Linked List.
9. Implementation of Queue using Linked List.
10. Implementation of Circular Queue using Linked List.
11. Implementation of Tree Structures, Binary Tree, Tree Traversal, Binary Search Tree, Insertion and Deletion in BST.
12. Graph Implementation, BFS, DFS, Minimum cost spanning tree, shortest path algorithm

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Database Management System (MCA-203)

Credit: 3+0

Course Objectives:

- To explain basic database concepts, applications, data models, schemas and instances.
- To demonstrate the use of constraints and relational algebra operations.
- To emphasize the importance of normalization in databases.
- To facilitate students in Database design.
- To familiarize issues of concurrency control and transaction management.

Syllabus

1. **Introduction:** Overview, Database System vs. File System, Database System Concept and Architecture, Data Model Schema and Instances, Data Independence and Database Language and Interfaces, Data Definitions Language, DML, Overall Database Structure. Data Modelling Using the Entity Relationship Model: ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Keys, Concepts of Super Key, Candidate Key, Primary Key, Generalization, Aggregation, Reduction of an ER Diagrams to Tables, Extended ER Model, Relationship of Higher Degree.

2. **Relational Data Model and Language:** Relational Data Model Concepts, Integrity Constraints, Entity Integrity, Referential Integrity, Keys Constraints, Domain Constraints, Relational Algebra, Relational Calculus, Tuple and Domain Calculus. Introduction on SQL: Characteristics of SQL, Advantage of SQL. SQl Data Type and Literals. Types of SQL Commands. SQL Operators and Their Procedure. Tables, Views and Indexes. Queries and Sub Queries. Aggregate Functions. Insert, Update and Delete Operations, Joins, Unions, Intersection, Minus, Cursors, Triggers, Procedures in SQL/PL SQL.

3. **Data Base Design & Normalization:** Functional dependencies, normal forms, first, second, 8 third normal forms, BCNF, inclusion dependence, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

4. **Transaction Processing Concept:** Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Checkpoints, Deadlock Handling. Distributed Database: Distributed Data Storage, Concurrency Control, Directory System.Concurrency Control Techniques: Concurrency Control, Locking Techniques for Concurrency Control, Time Stamping Protocols for Concurrency Control, Validation Based Protocol, Multiple Granularity, Multi Version Schemes, Recovery with Concurrent Transaction, Case Study of Oracle.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Apply knowledge of database for real life applications.

CO2: Apply query processing techniques to automate the real time problems of databases.

CO3: Identify and solve the redundancy problem in database tables using normalization.

CO4: Understand the concepts of transactions, their processing so they will familiar with broad range of database management issues including data integrity, security and recovery.

CO5: Design, develop and implement a small database project using database tools.

Text Books:

1. Korth, Silbertz, Sudarshan," Database Concepts", McGraw Hill.
2. Date C J, "An Introduction to Database Systems", Addision Wesley.
3. Elmasri, Navathe, "Fundamentals of Database Systems", Addision Wesley.
4. O'Neil, Databases, Elsevier Pub.
5. Ramakrishnan"Database Management Systems",McGraw Hill.
6. Leon & Leon,"Database Management Systems", Vikas Publishing House.
7. Bipin C. Desai, "An Introduction to Database Systems", Galgotia Publications.
8. Majumdar & Bhattacharya, "Database Management System", TMH.

Database Management System Lab (MCA-253)

Credit: 0+2

1. Installing oracle/ MYSQL
2. Creating Entity-Relationship Diagram using case tools.
3. Writing SQL statements Using ORACLE /MYSQL:
    a. Writing basic SQL SELECT statements.
    b. Restricting and sorting data.
    c. Displaying data from multiple tables.
    d. Aggregating data using group function.
    e. Manipulating data.
    f. Creating and managing tables.
4. Implementing Normalization
5. Creating cursor
6. Creating procedure and functions
7. Creating packages and triggers
8. Design and implementation of payroll processing system
9. Design and implementation of Library Information System
10. Design and implementation of Student Information System
11. Automatic Backup of Files and Recovery of Files
12. Sample of Mini project (Design & Development of Data and Application) for following:
    a. Inventory Control System.
    b. Material Requirement Processing.
    c. Hospital Management System.
    d. Railway Reservation System.
    e. Personal Information System.
    f. Web Based User Identification System.
    g. Timetable Management System.
    h. Hotel Management System

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner

Computer Based Numerical and Statistical Techniques (MCA-204)

Credit: 3+0

Course Objectives:

1. Develop an understanding of numerical accuracy, error analysis, and iterative methods for solving algebraic and transcendental equations.
2. Apply interpolation techniques, numerical differentiation, and integration methods for data approximation and solving differential equations.
3. Utilize statistical and regression techniques for data analysis, curve fitting, and quality control in computational applications.

Syllabus

1. **Introduction:** Numbers and their accuracy, Computer Arithmetic, Mathematical preliminaries, Errors and their Computation, General error formula, Error in a series approximation **Solution of Algebraic and Transcendental Equation:** Bisection Method, Iteration method, Method of false position, Newton-Raphson method, Methods of finding complex roots, Muller's method, Rate of convergence of Iterative methods, Polynomial Equations.
2. **Interpolation:** Finite Differences, Difference tables Polynomial Interpolation: Newton's forward and backward formula Central Difference Formulae: Gauss forward and backward formula, Stirling's, Bessel's, Everett's formula. Interpolation with unequal intervals: Langrange's Interpolation, Newton Divided difference formula, Hermite's Interpolation.
3. **Numerical Integration and Differentiation:** Introduction, Numerical differentiation Numerical Integration: Trapezoidal rule, Simpson's 1/3 and 3/8 rule, Boole's rule, Waddle's rule. **Solution of differential Equations:** Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta Methods, Predictor Corrector Methods, Automatic Error Monitoring and Stability of solution
4. **Statistical Computation:** Frequency chart, Curve fitting by method of least squares, fitting of straight lines, polynomials, exponential curves etc, Data fitting with Cubic splines, Regression Analysis, Linear and Nonlinear Regression, Multiple regression, Statistical Quality Control methods.

Course Outcome: After completion of this course, students will be able to

CO1: Understand numerical accuracy, computer arithmetic, error analysis, and iterative methods for solving algebraic and transcendental equations, including Bisection, Newton-Raphson, and Muller's methods.

CO2: Apply interpolation techniques such as Newton's forward and backward formulas, Gauss central difference formulas, Lagrange's interpolation, and Hermite's interpolation for numerical approximation.

CO3: Implement numerical differentiation and integration methods, including Trapezoidal Rule, Simpson's Rule, and Runge-Kutta Methods, for solving differential equations with stability analysis.

CO4: Utilize statistical computation techniques such as curve fitting, least squares method, cubic splines, and regression analysis (linear, nonlinear, and multiple) for data modeling and prediction.

CO5: Analyze and apply statistical quality control methods for data processing, error estimation, and optimization in computational and real-world applications.

References:

1. Rajaraman V, "Computer Oriented Numerical Methods", Pearson Education
2. Gerald & Whealey, "Applied Numerical Analyses", AW

Theory of Computation (MCA-205)

Credit: 3+0

Course Objective: This course focuses on the basic theory of Computer Science and formal methods of computation like automata theory, formal languages, grammars and Turing Machines. The objective of this course is to explore the theoretical foundations of computer science from the perspective of formal languages and classify machines by their power to recognize languages.

Syllabus

1. **Basic Concepts and Automata Theory:** Introduction to Theory of Computation- Automata, Computability and Complexity, Alphabet, Symbol, String, Formal Languages, Deterministic Finite Automaton (DFA)- Definition, Representation, Acceptability of a String and Language, Non Deterministic Finite Automaton (NFA), Equivalence of DFA and NFA, NFA with ε-Transition, Equivalence of NFA's with and without ε-Transition, Finite Automata with output- Moore Machine, Mealy Machine, Equivalence of Moore and Mealy Machine, Minimization of Finite Automata, Myhill-Nerode Theorem, Simulation of DFA and NFA.

2. **Regular Expressions and Languages:** Regular Expressions, Transition Graph, Kleen's Theorem, Finite Automata and Regular Expression- Arden's theorem, Algebraic Method Using Arden's Theorem, Regular and Non-Regular Languages- Closure properties of Regular Languages, Pigeonhole Principle, Pumping Lemma, Application of Pumping Lemma, Decidability- Decision properties, Finite Automata and Regular Languages, Regular Languages and Computers, Simulation of Transition Graph and Regular language.

3. **Regular and Non-Regular Grammars:** Context Free Grammar (CFG)-Definition, Derivations, Languages, Derivation Trees and Ambiguity, Regular Grammars-Right Linear and Left Linear grammars, Conversion of FA into CFG and Regular grammar into FA, Simplification of CFG, Normal Forms- Chomsky Normal Form (CNF), Greibach Normal Form (GNF), Chomsky Hierarchy, Programming problems based on the properties of CFGs.

4. **Push Down Automata and Properties of Context Free Languages:** Nondeterministic Pushdown Automata (NPDA)- Definition, Moves, A Language Accepted by NPDA, Deterministic Pushdown Automata (DPDA) and Deterministic Context free Languages (DCFL), Pushdown Automata for Context Free Languages, Context Free grammars for Pushdown Automata, two stack Pushdown Automata, Pumping Lemma for CFL, Closure properties of CFL, Decision Problems of CFL, Programming problems based on the properties of CFLs. Turing Machines and Recursive Function Theory: Basic Turing Machine Model, Representation of Turing Machines, Language Acceptability of Turing Machines, Techniques for Turing Machine Construction, Modifications of Turing Machine, Turing Machine as Computer of Integer Functions, Universal Turing machine, Linear Bounded Automata, Church's Thesis, Recursive and Recursively Enumerable language, Halting Problem, Post's Correspondence Problem, Introduction to Recursive Function Theory.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Analyse and design finite automata, pushdown automata, Turing machines, formal languages, and grammars.

CO2: Analyse and design, Turing machines, formal languages, and grammars.

CO3: Demonstrate the understanding of key notions, such as algorithm, computability, decidability, and complexity through problem solving.

CO4: Prove the basic results of the Theory of Computation.

CO5: State and explain the relevance of the Church-Turing thesis.

Text Books:

1. Introduction to Automata theory, Languages and Computation, J.E.Hopcraft, R.Motwani, and Ullman. 2nd edition, Pearson Education Asia
2. Introduction to languages and the theory of computation, J Martin, 3rd Edition, Tata McGraw Hill
3. Elements and Theory of Computation, C Papadimitrou and C. L. Lewis, PHI
4. Mathematical Foundation of Computer Science, Y.N.Singh, New Age International.

Advanced Computer Architecture (MCA-206)

Credit: 3+0

Course Objective: Conceptualize the basics of organizational and architectural issues of a digital computer like CPU, I/O interface and memory organization.

Syllabus

1. **Introduction:** Functional units of digital system and their interconnections, buses, bus architecture, types of buses and bus arbitration. Register, bus and memory transfer. Processor organization, general registers organization, stack organization and addressing modes.

2. **Arithmetic and Logic Unit:** Look ahead carries adders. Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Arithmetic & logic unit design. IEEE Standard for Floating Point Numbers. Control Unit: Instruction types, formats, instruction cycles and sub cycles (fetch and execute etc), micro-operations, execution of a complete instruction. Program Control, Reduced Instruction Set Computer, Pipelining. Hardwire and micro programmed control: micro programme sequencing, concept of horizontal and vertical microprogramming.

3. **Memory:** Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. ROM memories. Cache memories: concept and design issues & performance, address mapping and replacement Auxiliary memories: magnetic disk, magnetic tape and optical disks Virtual memory: concept implementation.

4. **Input / Output:** Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access., I/O channels and processors. Serial Communication: Synchronous & asynchronous communication, standard communication interfaces.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Study of the basic structure and operation of a digital computer system.

CO2: Analysis of the design of arithmetic & logic unit and understanding of the fixed point and floating-point arithmetic operations.

CO3: Implementation of control unit techniques and the concept of Pipelining.

CO4: Understanding the hierarchical memory system, cache memories and virtual memory.

CO5: Understanding the different ways of communicating with I/O devices and standard I/O interfaces>

Text Books:

1. Computer System Architecture - M. Mano,Pearson Publication
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012
3. John P. Hayes, Computer Architecture and Organization, Tata McGraw Hill, Third Edition, 1998.
4. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006.

5. Behrooz Parahami, "Computer Architecture", Oxford University Press, Eighth Impression, 2011.
6. David A. Patterson and John L. Hennessy, "Computer Architecture-A Quantitative Approach", Elsevier, a division of reed India Private Limited, Fifth edition, 2012
7. Structured Computer Organization, Tannenbaum (PHI)

Machine Learning (MCA-301)

Credit: 3+0

Course Objectives: This course provides an advanced level of understanding to machine learning and statistical pattern recognition. It offers some of the most cost-effective approaches to automated knowledge acquisition in emerging data-rich disciplines and focuses on the theoretical understanding of these methods, as well as their computational implications

Syllabus

1. **Introduction:** Learning, Types of Learning, well defined learning problems, Designing a Learning System, History of ML, Introduction of Machine Learning Approaches – (Artificial Neural Network, Clustering, Reinforcement Learning, Decision Tree Learning, Bayesian networks, Support Vector Machine, Genetic Algorithm), Issues in Machine Learning and Data Science Vs Machine Learning.

2. **Regression and Decision Tree Learning:** Linear Regression and Logistic Regression BAYESIAN LEARNING - Bayes theorem, Concept learning, Bayes Optimal Classifier, Naïve Bayes classifier, Bayesian belief networks, EM algorithm. SUPPORT VECTOR MACHINE: Introduction, Types of support vector kernel – (Linear kernel, polynomial kernel, and Gaussian kernel), Hyperplane – (Decision surface), Properties of SVM, and Issues in SVM. DECISION TREE LEARNING: Decision tree learning algorithm, Inductive bias, Inductive inference with decision trees, Entropy and information theory, Information gain, ID-3 Algorithm, Issues in Decision tree learning. INSTANCE-BASED LEARNING – k-Nearest Neighbour Learning, Locally Weighted Regression, Radial basis function networks, Case-based learning.

3. **Artificial Neural Networks:** Perceptron's, Multilayer perceptron, Gradient descent and the Delta rule, Multilayer networks, Derivation of Backpropagation Algorithm, Generalization, Unsupervised Learning – SOM Algorithm and its variant; DEEP LEARNING - Introduction, concept of convolutional neural network , Types of layers – (Convolutional Layers , Activation function , pooling , fully connected) , Concept of Convolution (1D and 2D) layers, Training of network, Case study of CNN for e.g. on Diabetic Retinopathy, Building a smart speaker, Self-deriving car etc.

4. **Reinforcement Learning:** Introduction to Reinforcement Learning, Learning Task, Example of Reinforcement Learning in Practice, Learning Models for Reinforcement – (Markov Decision process, Q Learning - Q Learning function, Q Learning Algorithm), Application of Reinforcement Learning, Introduction to Deep Q Learning. GENETIC ALGORITHMS: Introduction, Components, GA cycle of reproduction, Crossover, Mutation, Genetic Programming, Models of Evolution and Learning, Applications.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Understand the need for machine learning for various problem solving.

CO2: Understand a wide variety of learning algorithms and how to evaluate models generated from data.

CO3: Understand the latest trends in machine learning.

CO4: Design appropriate machine learning algorithms and apply the algorithms to a real-world problem.

CO5: Optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Text Books:

1. Tom M. Mitchell, ―Machine Learning, McGraw-Hill Education (India) Private Limited, 2013.
2. Ethem Alpaydin, ―Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press 2004.
3. Stephen Marsland, ―Machine Learning: An Algorithmic Perspective, CRC Press, 2009.
4. Bishop, C., Pattern Recognition and Machine Learning. Berlin: Springer-Verlag.

Computer Network (MCA-302)

Credit: 3+0

Course Objective: This course studies the standard models for the layered approach to communication between autonomous machines in a network, and the main characteristics of data transmission across various physical link types. It considers how to design networks and protocols for diverse situations, analyses several application and support protocols from a distributed systems viewpoint, and identifies significant problem areas in networked communications.

Syllabus

1. **Introductory Concepts:** Goals and applications of networks, Categories of networks, Organization of the Internet, ISP, Network structure and architecture (layering principles, services, protocols and standards), The OSI reference model, TCP/IP protocol suite, Network devices and components. Physical Layer: Network topology design, Types of connections, Transmission media, Signal transmission and encoding, Network performance and transmission impairments, switching techniques and multiplexing.

2. **Link Layer:** Framing, Error Detection and Correction, Flow control (Elementary Data Link Protocols, Sliding Window protocols). Medium Access Control and Local Area Networks: Channel allocation, multiple access protocols, LAN standards, Link layer switches & bridges (learning bridge and spanning tree algorithms).

3. **Network Layer:** Point-to-point networks, Logical addressing, Basic internetworking (IP, CIDR, ARP, RARP, DHCP, and ICMP), Routing, forwarding and delivery, Static and dynamic routing, Routing algorithms and protocols, Congestion control algorithms, IPv6.

4. **Transport Layer:** Process-to-process delivery, Transport layer protocols (UDP and TCP), Multiplexing, Connection management, Flow control and retransmission, Window management, TCP Congestion control, Quality of service. **Application Layer:** Domain Name System, World Wide Web and Hyper Text Transfer Protocol, Electronic mail, File Transfer Protocol, Remote login, Network management, Data compression, Cryptography – basic concepts.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Explain basic concepts, OSI reference model, services and role of each layer of OSI model and TCP/IP, networks devices and transmission media, Analog and digital data transmission.

CO2: Apply channel allocation, framing, error and flow control techniques.

CO3: Describe the functions of Network Layer i.e., Logical addressing, subnetting & Routing Mechanism.

CO4: Explain the different Transport Layer function i.e., Port addressing, Connection Management, Error control and Flow control mechanism.

CO5: Explain the functions offered by session and presentation layer and their Implementation.

CO6: Explain the different protocols used at application layer i.e., HTTP, SNMP, SMTP, FTP, TELNET and VPN.

Text Books:

1. Behrouz Forouzan, "Data Communication and Networking", McGraw Hill
2. Andrew Tanenbaum "Computer Networks", Prentice Hall.
3. William Stallings, "Data and Computer Communication", Pearson.
4. Kurose and Ross, "Computer Networking- A Top-Down Approach", Pearson.
5. Peterson and Davie, "Computer Networks: A Systems Approach", Morgan Kaufmann
6. W. A. Shay, "Understanding Communications and Networks", Cengage Learning.
7. D. Comer, "Computer Networks and Internets", Pearson.
8. Behrouz Forouzan, "TCP/IP Protocol Suite", McGraw Hill.

Design and Analysis of Algorithms (MCA-303)

Credit: 3+0

Course Objective: The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able to develop their own versions for a given computational task and to compare and contrast their performance.

Syllabus

1. Introduction: Algorithms, Analyzing Algorithms, Complexity of Algorithms, Growth of Functions, Performance Measurements, Sorting and Order Statistics - Shell Sort, Quick Sort, Merge Sort, Heap Sort, Comparison of Sorting Algorithms, Sorting in Linear Time.

2. Advanced Data Structures: Red-Black Trees, B – Trees, Binomial Heaps, Fibonacci Heaps, Tries, Skip List.

3. Divide and Conquer with Examples Such as Sorting, Matrix Multiplication, Convex Hull and Searching. Greedy Methods with Examples Such as Optimal Reliability Allocation, Knapsack, Minimum Spanning Trees – Prim's and Kruskal's Algorithms, Single Source Shortest Paths - Dijkstra's and Bellman Ford Algorithms.

4. Dynamic Programming with Examples Such as Knapsack. All Pair Shortest Paths – Warshal's and Floyd's Algorithms, Resource Allocation Problem. Backtracking, Branch and Bound with Examples Such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of Subsets.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Design new algorithms, prove them correct, and analyse their asymptotic and absolute runtime and memory demands.

CO2: Find an algorithm to solve the problem (create) and prove that the algorithm solves the problem correctly (validate).

CO3: Understand the mathematical criterion for deciding whether an algorithm is efficient, and know many practically important problems that do not admit any efficient algorithms.

CO4: Apply classical sorting, searching, optimization and graph algorithms.

CO5: Understand basic techniques for designing algorithms, including the techniques of recursion, divide-and-conquer, and greedy.

Text Books:

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms", Printice Hall of India.
2. E. Horowitz & S Sahni, "Fundamentals of Computer Algorithms",

3. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.
4. LEE "Design & Analysis of Algorithms (POD)", McGraw Hill
5. Richard E.Neapolitan "Foundations of Algorithms" Jones & Bartlett Learning
6. Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
7. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples,Second Edition, Wiley, 2006.
8. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins, 1997
9. Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.
10. Harsh Bhasin,"Algorithm Design and Analysis",First Edition,Oxford University Press.
11. Gilles Brassard and Paul Bratley,Algorithmics:Theory and Practice,Prentice Hall,1995.

Distributed System (MCA-304)

Credit: 3+0

Course Objective: This course is an introduction to the design of distributed systems and algorithms that support distributed computing.

Syllabus

1. **Characterization of Distributed Systems:** Introduction, Examples of distributed Systems, Resource sharing and the Web Challenges. Architectural models, Fundamental Models. Theoretical Foundation for Distributed System: Limitation of Distributed system, absence of global clock, shared memory, Logical clocks, Lamport's & vectors logical clocks. Concepts in Message Passing Systems: causal order, total order, total causal order, Techniques for Message Ordering, Causal ordering of messages, global state, and termination detection.

2. **Distributed Mutual Exclusion:** Classification of distributed mutual exclusion, requirement of mutual exclusion theorem, Token based and non-token-based algorithms, performance metric for distributed mutual exclusion algorithms. Distributed Deadlock Detection: system model, resource Vs communication deadlocks, deadlock prevention, avoidance, detection & resolution, centralized dead lock detection, distributed dead lock detection, path pushing algorithms, edge chasing algorithms.

3. **Agreement Protocols:** Introduction, System models, classification of Agreement Problem, Byzantine agreement problem, Consensus problem, Interactive consistency Problem, Solution to Byzantine Agreement problem, Application of Agreement problem, Atomic Commit in Distributed Database system. Distributed Resource Management: Issues in distributed File Systems, Mechanism for building distributed file systems, Design issues in Distributed Shared Memory, Algorithm for Implementation of Distributed Shared Memory.

4. **Failure Recovery in Distributed Systems & Transactions and Concurrency Control:** Concepts in Backward and Forward recovery, Recovery in Concurrent systems, obtaining consistent Checkpoints, Recovery in Distributed Database Systems. Fault Tolerance: Issues in Fault Tolerance, Commit Protocols, Voting protocols, Dynamic voting protocols. **Transactions and Concurrency Control:** Transactions, Nested transactions, Locks, Optimistic Concurrency control, Timestamp ordering, Comparison of methods for concurrency control. Distributed Transactions: Flat and nested distributed transactions, Atomic Commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Replication: System model and group communication, Fault - tolerant services, highly available services, Transactions with replicated data.

Course Outcome (CO) : At the end of course, the student will be able to

CO1: Provide hardware and software issues in modern distributed systems.

CO2: Get knowledge in distributed architecture, naming, synchronization, consistency and replication, fault tolerance, security, and distributed file systems.

CO3: Analyze the current popular distributed systems such as peer-to-peer (P2P) systems will also be analyzed.

CO4: Know about Shared Memory Techniques and have sufficient knowledge about file access.

CO5: Have knowledge of Synchronization and Deadlock.

Text Books:

1. Singhal & Shivaratri, "Advanced Concept in Operating Systems", McGraw Hill
2. Ramakrishna,Gehrke," Database Management Systems", McGraw Hill
3. Vijay K.Garg Elements of Distributed Computing , Wiley
4. Coulouris, Dollimore, Kindberg, "Distributed System: Concepts and Design", Pearson Education

5.Tenanuanbaum, Steen," Distributed Systems", PHI

Professional Elective Course

(Offered by Department of Computer Science and Engineering)

Professional Elective Course List-1

Introduction to Software Testing (MCAE-301)

Credit: 3+0

Course Objective:

- To study fundamental concepts in software testing
- To discuss various software testing issues and solutions in software unit test, integration and system testing.
- To expose the advanced software testing topics, such as object-oriented software testing methods.

Syllabus

1. **Review of Software Engineering:** Overview of Software Evolution, SDLC, Testing Process, Terminologies in Testing: Error, Fault, Failure, Verification, Validation, Difference Between Verification and Validation, Test Cases, Testing Suite, Test, Oracles, Impracticality of Testing All Data; Impracticality of Testing All Paths. Verification: Verification Methods, SRS Verification, Source Code Reviews, User Documentation Verification, Software, Project Audit, Tailoring Software Quality Assurance Program by Reviews, Walkthrough, Inspection and Configuration Audits.

2. **Functional Testing:** Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique. Structural Testing: Control Flow Testing, Path Testing, Independent Paths, Generation of Graph from Program, Identification of Independent Paths, Cyclomatic Complexity, Data Flow Testing, Mutation Testing.

3. **Regression Testing:** What is Regression Testing? Regression Test cases selection, Reducing the number of test cases, Code coverage prioritization technique. Reducing the number of test cases: Prioritization guidelines, Priority category, Scheme, Risk Analysis.

4. **Software Testing Activities & Object-Oriented Testing:** Levels of Testing, Debugging, Testing techniques and their applicability, Exploratory Testing Automated Test Data Generation: Test Data, Approaches to test data generation, test data generation using genetic algorithm, Test Data Generation Tools, Software Testing Tools, and Software test Plan. **Object Oriented Testing:** Definition, Issues, Class Testing, Object Oriented Integration and System Testing. Testing Web Applications: Web Testing, User Interface Testing, Usability Testing, Security Testing, Performance Testing, Database testing, Post Deployment Testing.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Have an ability to apply software testing knowledge and engineering methods.

CO2: Have an ability to design and conduct a software test process for a software testing project.

CO3: Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.

CO4: Have an ability understand and identify various software testing problems, and solve these Problems by designing and selecting software test models, criteria, strategies, and methods.

CO5: Have basic understanding and knowledge of contemporary issues in software testing, such as Component-based software testing problems.

Text Books:

1. Yogesh Singh, "Software Testing", Cambridge University Press, New York, 2012
2. K.K. Aggarwal & Yogesh Singh, "Software Engineering", New Age International Publishers, New Delhi, 2003.
3. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", Fifth Edition, McGraw-Hill International Edition, New Delhi,2001.
4. Marc Roper, "Software Testing", McGraw-Hill Book Co., London, 1994.
5. M.C. Trivedi, Software Testing & Audit, Khanna Publishing House 6. Boris Beizer, "Software System

Testing and Quality Assurance", Van Nostrand Reinhold, New York, 1984

Wireless Network Systems (MCAE-302)

Credit: 3+0

Course Objective:

- To study the evolving wireless technologies and standards
- To understand the architectures of various access technologies such as 3G, 4G.
- To understand various protocols and services provided by next generation networks.

Syllabus

1. **Introduction to WLAN Technologies (IEEE802.11):** System architecture, protocol architecture, 802.11b, 802.11a – Hiper LAN: WATM, BRAN, HiperLAN2 – Bluetooth: Architecture, WPAN – IEEE 802.15.4, Wireless USB, Zigbee, 6LoWPAN, Wireless HART.**MOBILE NETWORK LAYER** Introduction to Mobile IP, IP packet delivery, Agent discovery, tunneling and encapsulation, IPV6- Network layer in the internet- Mobile IP session initiation protocol - mobile ad-hoc network: Routing: Destination Sequence distance vector, IoT: CoAP

2. **3G Overview** of UTMS Terrestrial Radio access network-UMTS Core network Architecture: 3GPP Architecture, User equipment, CDMA2000 overview- Radio and Network components, Network structure, Radio Network, TD-CDMA, TD – SCDMA.

3. **Internetworking between WLANS and WWANS** Internetworking objectives and requirements, Schemes to connect WLANS and 3G Networks, Session Mobility, Internetworking Architecture for WLAN and GPRS, System Description, Local Multipoint Distribution Service, Multichannel Multipoint Distribution System.

4. **4G & Beyond** Introduction – 4G vision – 4G features and challenges - Applications of 4G – 4G Technologies: Multicarrier Modulation, Smart antenna techniques, IMS Architecture, LTE, Advanced Broadband Wireless Access and Services, MVNO.

Course Outcome (CO) : Upon completion of the course, the student would be able to

CO1: Conversant with the latest 3G/4G networks and its architecture

CO2: Design and implement wireless network environment for any application using latest wireless protocols and standards

CO3: Ability to select the suitable network depending on the availability and requirement

CO4: Implement different type of applications for smart phones and mobile devices with latest network strategies

Text Books:

1. Jochen Schiller, Mobile Communications, Second Edition, Pearson Education 2012.
2. Vijay Garg, —Wireless Communications and networking, First Edition, Elsevier 2007.
3. Erik Dahlman, Stefan Parkvall, Johan Skold and Per Beming, "3G Evolution HSPA and LTE for Mobile Broadband, Second Edition, Academic Press, 2008.
4. Anurag Kumar, D.Manjunath, Joy kuri, —Wireless Networking, First Edition, Elsevier 2011.
5. Simon Haykin , Michael Moher, David Koilpillai, —Modern Wireless Communications, First Edition, Pearson Education 2013

Artificial Intelligence (MCAE-303)

Credit: 3+0

Course Objective: This course sheds light on the fundamental of Artificial Intelligence and its applications in various areas.

Syllabus

1. **Introduction:** Definition – Future of Artificial Intelligence – Characteristics of Intelligent Agents– Typical Intelligent Agents – Problem Solving Approach to Typical AI problems.

2. **Problem Solving Methods:** Problem solving Methods – Search Strategies- Uninformed – Informed – Heuristics – Local Search Algorithms and Optimization Problems – Searching with Partial Observations – Constraint Satisfaction Problems – Constraint Propagation – Backtracking Search – Game Playing – Optimal Decisions in Games – Alpha – Beta Pruning – Stochastic Games.

3. **Knowledge Representation:** First Order Predicate Logic – Prolog Programming – Unification – Forward Chaining-Backward Chaining – Resolution – Knowledge Representation – Ontological Engineering-Categories and Objects – Events – Mental Events and Mental Objects – Reasoning Systems for Categories – Reasoning with Default Information.

4. **Software Agents & Applications:** Architecture for Intelligent Agents – Agent communication – Negotiation and Bargaining – Argumentation among Agents – Trust and Reputation in Multi-agent systems. **Applications:** AI applications – Language Models – Information Retrieval-Information Extraction – Natural Language Processing – Machine Translation – Speech Recognition – Robot – Hardware – Perception – Planning – Moving.

Course Outcome (CO): At the end of course, the student will be able

CO1: To understand the basics of the theory and practice of Artificial Intelligence as a discipline and about intelligent agents.

CO2: Understand search techniques and gaming theory.

CO3: The student will learn to apply knowledge representation techniques and problem-solving strategies to common AI applications.

CO4: Student should be aware of techniques used for classification and clustering.

CO5: Student should aware of basics of pattern recognition and steps required for it.

Text Books:

1. S. Russell and P. Norvig, “Artificial Intelligence: A Modern Approach‖, Prentice Hall, Third Edition, 2009.
2. Bratko, “Prolog: Programming for Artificial Intelligence”, Fourth edition, Addison-Wesley Educational Publishers Inc., 2011.
3. M. Tim Jones, —Artificial Intelligence: A Systems Approach, Jones and Bartlett Publishers, Inc.First Edition, 2008
4. Nils J. Nilsson, —The Quest for Artificial Intelligence, Cambridge University Press, 2009.
5. William F. Clocksin and Christopher S. Mellish, Programming in Prolog: Using the ISO Standard, Fifth Edition, Springer, 2003.

6. Gerhard Weiss, ―Multi Agent Systems, Second Edition, MIT Press, 2013.
7. David L. Poole and Alan K. Mackworth, ―Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, 2010.

Data Mining and Data Warehousing (MCAE-304)

Credit: 3+0

Course Objective:

- The course content enables students to:
- Analyze the difference between On Line Transaction Processing and On-Line analytical processing.
- Create Multidimensional schemas suitable for data warehousing.
- Understand various data mining functionalities.
- Understand in detail about data mining algorithms.

Syllabus

1. **Data Warehousing and Business Analysis:** Data warehousing Components –Building a Data warehouse –Data Warehouse Architecture – DBMS Schemas for Decision Support – Data Extraction, Cleanup, and Transformation Tools –Metadata – reporting – Query tools and Applications – Online Analytical Processing (OLAP) – OLAP and Multidimensional Data Analysis.

2. **Data Mining:** Data Mining Functionalities – Data Preprocessing – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization and Concept Hierarchy Generation- Architecture of A Typical Data Mining Systems- Classification of Data Mining Systems. **Association Rule Mining:** - Efficient and Scalable Frequent Item set Mining Methods – Mining Various Kinds of Association Rules – Association Mining to Correlation Analysis – Constraint-Based Association Mining.

3. **Classification and Prediction:** Issues Regarding Classification and Prediction – Classification by Decision Tree Introduction – Bayesian Classification – Rule Based Classification – Classification by Back propagation – Support Vector Machines – Associative Classification – Lazy Learners – Other Classification Methods – Prediction – Accuracy and Error Measures – Evaluating the Accuracy of a Classifier or Predictor – Ensemble Methods – Model Section.

4. **Cluster Analysis, Mining Object, Spatial, Multimedia, Text and Web Data:** Types of Data in Cluster Analysis – A Categorization of Major Clustering Methods – Partitioning Methods – Hierarchical methods – Density-Based Methods – Grid-Based Methods – Model-Based Clustering Methods – Clustering High-Dimensional Data – Constraint-Based Cluster Analysis – Outlier Analysis. **Mining Object, Spatial, Multimedia, Text and Web Data:** Multidimensional Analysis and Descriptive Mining of Complex Data Objects – Spatial Data Mining – Multimedia Data Mining – Text Mining – Mining the World Wide Web.

Course Outcome (CO): At the end of course, the student will be able to

CO1: To understand the principles of Data Warehousing and Data Mining.

CO2: To be familiar with the Data warehouse architecture and its Implementation.

CO3: To know the Architecture of a Data Mining system.

CO4: To understand the various Data preprocessing Methods.

CO5: To perform classification and prediction of data.

Text Books:

1. Jiawei Han, Micheline Kamber and Jian Pei "Data Mining Concepts and Techniques", Third Edition, Elsevier, 2011.
2. Alex Berson and Stephen J. Smith "Data Warehousing, Data Mining & OLAP", Tata McGraw – Hill Edition, Tenth Reprint 2007.

3. K.P. Soman, Shyam Diwakar and V. Ajay "Insight into Data mining Theory and Practice", Easter Economy Edition, Prentice Hall of India, 2006.
4. G. K. Gupta "Introduction to Data Mining with Case Studies", Easter Economy Edition, Prentice Hall of India, 2006.
5. Pang-Ning Tan, Michael Steinbach and Vipin Kumar "Introduction to Data Mining", Pearson.

Advanced Database Management System (MCAE-305)

Credit: 3+0

Course Objectives:

- To provide a strong foundation in advanced database concepts from an industry perspective.
- To covers advanced data modeling concepts like OOD Modeling and ORD Modeling
- To learn query processing and transaction management concepts for object-relational database and distributed database

Syllabus

1. Transaction and schedules, Concurrent Execution of transaction, Conflict and View Serializability, Testing for Serializability, Concepts in Recoverable and Cascade less schedules. Lock based protocols, time stamp-based protocols, Multiple Granularity and Multiversion Techniques, enforcing serializablity by Locks, Locking system with multiple lock modes, architecture for Locking scheduler

2. Distributed Transactions Management, Data Distribution,fragmentation and Replication Techniques, Distributed Commit, Distributed Locking schemes, Long duration transactions,Moss Concurrency protocol.

3. Issues of Recovery and atomicity in Distributed Databases, Traditional recovery techniques, Log based recovery, Recovery with Concurrent Transactions, Recovery in Message passing systems, Checkpoints, Algorithms for recovery line, Concepts in Orphan and Inconsistent Messages.

4. Distributed Query Processing, Multiway Joins, Semi joins, Cost based query optimization for distributed database, Updating replicated data, protocols for Distributed Deadlock Detection, Eager and Lazy Replication Techniques

Course Outcome (CO): At the end of course, the student will be able

CO1: To understand the principles of transaction and schedules.

CO2: To be familiar with the concepts of distributed transaction management.

CO3: To know the issues of recovery in transactions.

CO4: To understand the issues of the concurrent transaction processing.

CO5: To know about distributed query processing.

Text Books:

1. Silberschatz, Korth and Sudershan, Database System Concept', Mc Graw Hill
2. Ramakrishna and Gehrke,' Database Management System, Mc Graw Hill
3. Garcia-Molina, Ullman,Widom,' Database System Implementation' Pearson Education 2. Ceei and Pelagatti,'Distributed Database', TMH
4. Singhal and Shivratri, 'Advance Concepts in Operating Systems' MC Graw Hill

Design & Development of Mobile Applications (MCAE-306)

Credit: 3+0

Course Objective: A course in Design & Development of Applications with an emphasis on software development. Integrated development tools, software development kits, and software subsystems are to be employed to develop database, eCommerce, and mobile applications.

Syllabus

1. **Introduction:** Introduction to mobile applications – Embedded systems - Market and business drivers for mobile applications – Publishing and delivery of mobile applications – Requirements gathering and validation for mobile applications.

2. **Basic Design:** Introduction – Basics of embedded systems design – Embedded OS – Design constraints for mobile applications, both hardware and software related – Architecting mobile applications – User interfaces for mobile applications – touch events and gestures – Achieving quality constraints – performance, usability, security, availability and modifiability.

3. **Advanced Design:** Designing applications with multimedia and web access capabilities – Integration with GPS and social media networking applications – Accessing applications hosted in a cloud computing environment – Design patterns for mobile applications.

4. **Introduction to Android And Objective C:** Establishing the development environment – Android architecture – Activities and views – Interacting with UI – Persisting data using SQLite – Packaging and deployment – Interaction with server-side applications – Using Google Maps, GPS and Wi-Fi – Integration with social media applications. **INTRODUCTION TO OBJECTIVE C:** iOS features, UI implementation, Touch frameworks – Data persistence using Core Data and SQLite – Location aware applications using Core Location and Map Kit – Integrating calendar and address book with social media application – Using Wi-Fi - iPhone marketplace. Swift: Introduction to Swift, features of swift.

Course Outcome (CO): At the end of course, the student will be

CO1: Able to expose the technology and business trends impacting mobile applications.

CO2: Competent with the characterization and architecture of mobile applications.

CO3: Competent with understanding enterprise scale requirements of mobile applications.

CO4: Competent with designing and developing mobile applications using one application development framework.

CO5: Able to expose the Android and iOS platforms to develop the mobile applications

Text Books:

1. Charlie Collins, Michael Galpin and Matthias Kappler, “Android in Practice”, DreamTech, 2012
2. AnubhavPradhan , Anil V Despande Composing Mobile Apps,Learn ,explore,apply
3. James Dovey and Ash Furrow, “Beginning Objective C”, Apress, 2012
4. Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", Wrox, 2012
5. David Mark, Jack Nutting, Jeff LaMarche and Frederic Olsson, “Beginning iOS
6. Development: Exploring the iOS SDK”, Apress, 2013.

Design & Analysis of Parallel Algorithms (MCAE-307)

Credit: 3+0

Course Objective: This module introduces theoretical design principles and analysis techniques that enable the creation and evaluation of efficient, scalable and portable algorithms for parallel computers. Concrete examples will span a range of application areas and architectural models seeking wherever possible to exploit commonality through appropriate abstraction.

Syllabus

1. **Introduction:** Sequential model, need of alternative model, parallel computational models such as PRAM, LMCC, Hypercube, Cube Connected Cycle, Butterfly, Perfect Shuffle Computers, Tree model, Pyramid model, Fully Connected model, PRAM-CREW, EREW models, simulation of one model from another one.

2. **Performance Measures of Parallel Algorithms:** speed-up and efficiency of PA, Cost-optimality, an example of illustrate Cost- optimal algorithms- such as summation, Min/Max on various models. **Parallel Sorting Networks:** Parallel Merging Algorithms on CREW/EREW/MCC, Parallel Sorting Networks CREW/EREW/MCC/, linear array.

3. **Parallel Searching Algorithm:** Kth element, Kth element in X+Y on PRAM, Parallel Matrix Transportation and Multiplication Algorithm on PRAM, MCC, Vector-Matrix Multiplication, Solution of Linear Equation, Root finding.

4. **Graph Algorithms:** Connected Graphs, search and traversal, Combinatorial Algorithms-Permutation, Combinations, Derrangements.

Course Outcome (CO): At the end of course, the student will be able

CO1: To know about the models of parallel computing.

CO2: To be familiar with the parallel sorting networks.

CO3: To know about parallel searching algorithms.

CO4: To understand the issues of graph algorithms in parallel algorithms.

Text Books:

1. M.J. Quinn, “Designing Efficient Algorithms for Parallel Computer”, McGrawHill.
2. S.G. Akl, “Design and Analysis of Parallel Algorithms”, by Academic Press
3. S.G. Akl,” Parallel Sorting Algorithm” by Academic Press

High Performance Computing (MCAE-308)

Credit: 3+0

Course Objective: The objective of this course is to teach participants the basic concepts of parallel computing and equip them with knowledge which will be sufficient enough for them to write parallel programs using industry-standard parallel programming frameworks.

Syllabus

1. **Overview of Grid Computing Technology& Open Grid Services Architecture:** History of Grid Computing, High Performance Computing, Cluster Computing. Peer-to-Peer Computing, Internet Computing, Grid Computing Model and Protocols, Types of Grids: Desktop Grids, Cluster Grids, Data Grids, High- Performance Grids, Applications and Architectures of High-Performance Grids, High Performance Application Development Environment. **Open Grid Services Architecture:** Introduction, Requirements, Capabilities, Security Considerations, GLOBUS Toolkit

2. **Overview of Cluster Computing:** Cluster Computer and its Architecture, Clusters Classifications, Components for Clusters, Cluster Middleware and SSI, Resource Management and Scheduling, Programming, Environments and Tools, Cluster Applications, Cluster Systems,

3. **Beowulf Cluster:** The Beowulf Model, Application Domains, Beowulf System Architecture, Software Practices, Parallel Programming with MPL, Parallel Virtual Machine (PVM).

4. **Overview of Cloud Computing:** Types of Cloud, Cyber infrastructure, Service Oriented Architecture Cloud Computing Components: Infrastructure, Storage, Platform, Application, Services, Clients, Cloud Computing Architecture.

Course Outcome (CO) : At the end of course, the student will be able

CO1: To understand the basic concept of Computer architecture and Modern Processor.

CO2: To understand the basic concepts of access optimization and parallel computers.

CO3: To describe different parallel processing platforms involved in achieving high performance computing.

CO4: Develop efficient and high-performance parallel programming.

CO5: Able to learn parallel programming using message passing paradigm.

Text Books:

1. Laurence T.Yang, Minyi Guo – High Performance Computing Paradigm and Infrastructure John Wiley
2. Ahmar Abbas, "Grid Computing: Practical Guide to Technology & Applications", Firewall Media, 2004.
3. Joshy Joseph and Craig Fellenstein , "Grid Computing" Pearson Education, 2004.
4. lan Foster, et al.,"The Open Grid Services Architecture", Version 1.5 (GFD.80). Open Grid Forum, 2006.
5. RajkumarBuyya. High Performance Cluster Computing: Architectures and Systems. Prentice Hall India, 1999.

Professional Elective Course List-2

Compiler Design (MCAE-401)

Credit: 3+0

Course Objective: The main objective of this course is to introduce the major concept areas of language translation and compiler design and to develop an awareness of the function and complexity of modern compilers. This course is a study of the theory and practice required for the design and implementation of interpreters and compilers for programming languages.

Syllabus

1. **Introduction to Compiler:** Phases and passes, Bootstrapping, Finite state machines and regular expressions and their applications to lexical analysis, Optimization of DFA-Based Pattern Matchers implementation of lexical analyzers, lexical-analyzer generator, LEX compiler, Formal grammars and their application to syntax analysis, BNF notation, ambiguity, YACC. The syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilities of CFG.

2. **Basic Parsing Techniques:** Parsers, Shift reduce parsing, operator precedence parsing, top-down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR (0) items, constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.

3. **Syntax-directed Translation:** Syntax-directed Translation schemes, Implementation of Syntax-directed Translators, Intermediate code, postfix notation, Parse trees & syntax trees, three address code, quadruple & triples, translation of assignment statements, Boolean expressions, statements that alter the flow of control, postfix translation, translation with a top-down parser. More about translation: Array references in arithmetic expressions, procedures call, declarations and case statements.

4. **Symbol Tables & Code Generation:** Data structure for symbols tables, representing scope information. Run-Time Administration: Implementation of simple stack allocation scheme, storage allocation in block structured language. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors semantic errors. Code Generation: Design Issues, the Target Language. Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Code Generator. Code optimization: Machine-Independent Optimizations, Loop optimization, DAG representation of basic blocks, value numbers and algebraic laws, Global Data-Flow analysis.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Acquire knowledge of different phases and passes of the compiler and also able to use the compiler tools like LEX, YACC, etc. Students will also be able to design different types of compiler tools to meet the requirements of the realistic constraints of compilers.

CO2: Understand the parser and its types i.e., Top-Down and Bottom-up parsers and construction of LL, SLR, CLR, and LALR parsing table.

CO3: Implement the compiler using syntax-directed translation method and get knowledge about the synthesized and inherited attributes.

CO4: Acquire knowledge about run time data structure like symbol table organization and different techniques used in that.

CO5: Understand the target machine's run time environment, its instruction set for code generation and techniques used for code optimization.

Text Books:

1. Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools", Pearson Education
2. K. Muneeswaran,Compiler Design,First Edition,Oxford University Press
3. J.P. Bennet, "Introduction to Compiler Techniques", Second Edition, McGraw-Hill,2003.
4. Henk Alblas and Albert Nymeyer, "Practice and Principles of Compiler Building with C", PHI, 2001.
5. V Raghvan, "Principles of Compiler Design", McGraw-Hill.
6. Kenneth Louden," Compiler Construction", Cengage Learning.
7. Charles Fischer and Ricard LeBlanc," Crafting a Compiler with C", Pearson Education

Back-End Technologies (MCAE-402)

Credit: 3+0

Course Objective:

- To learn the basics of Angular JS
- To implement Forms, inputs and Services
- To implement Directives and Databases
- To understand basics of Nodejs and Express
- To understand the basics of Mongo DB and its data model

Syllabus

1. **Introduction to Angular JS:** Introduction – Features – Angular JSModel-View-Controller – Expression -Directives and Controllers - Angular JS Modules – Arrays –Working with ng-model – Working with Forms – Form Validation – Error Handling with Forms – Nested Forms with ng-form – Other Form Controls.

2. **Directives & Building Databases:** Filters – Using Filters in Controllers and Services – Angular JS Services – Internal Angular JS Services – Custom Angular JS Services - Directives – Alternatives to Custom Directives – Understanding the Basic options – Interacting with Server –HTTP Services – Building Database, Front End and Back End.

3. **Node JS and Express Framework:** Introduction –Using the Terminals – Editors –Building a Webserver with Node – The HTTP Module – Views and Layouts –Middle ware – Routing – Form Handling with Express - The Request and Response Objects –Handle bars – Comments and Blocks.

4. **Introduction to MONGODB:** JSON and MongoDB – Adopting a Non-relational Approach – Opting for Performance vs. Features Running the Database Anywhere – Generating or Creating a Key – Using Keys and Values – Implementing Collections **DATA MODELS:** Understanding Databases – Reviewing the Feature List - Designing the Database – Building Indexes – Inserting Data – Quering for Data – Updating Data – Removing Data – Referencing a Database – GRID FS.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: Implement Directives and Controllers.

CO2: Create Forms, validate and use Filters.

CO3: Understand the usage of Keys and Values.

CO4: Implementations of data model.

Text Books:

1. Adam Freeman - ProAngular JS, Apress, First Edition, 2014.
2. ShyamSeshadri, Brad Green –"AngularJS: Up and Running: Enhanced Productivity with tructured Web Apps", Apress, O'Reilly Media, Inc.
3. AgusKurniawan–"AngularJS Programming by Example", First Edition, PE Press, 2014.
4. David Hows, Peter Membrey, EelcoPlugge – "MongoDB Basics", Apress, 2014.
5. Ethan Brown, "Web Development with Node and Express", Oreilly Publishers, First Edition.

Front End Technologies (MCAE-403)

Credit: 3+0

Course Objective:

- To provide an introduction to the principles and practices of Network Engineering.
- To understand the architecture of the network devices.
- To learn QoS related methodologies.
- To explore the emerging technologies in network engineering.

Syllabus

1. **HTML5 & CSS3 –** Introduction, Elements, Tags, Lists, Tables, Images, Forms - Form Elements & Attributes, Hidden Fields, Semantic Elements, Media Elements, Canvas, SVG, Drag & Drop, Geolocation, Web Storage, Special Tags, Formatting Tags. CSS – Introduction, Styling, Box Model, Padding & Dimension, Transforms, Transitions, Animations, Multiple columns, User Interface

2. **Bootstrap -** Overview, Environment setup, Precompiled Bootstrap, Source Code, Grid System, Bootstrap CSS Overview, Typography, Code, Tables, Forms, Helper Classes, Responsive Utilities, Glyphicons, Dropdowns, Navigation Elements, Breadcrumb, Pagination, Badges, Progress bars, Plugins Overview, Transition Plugin, Model Plugin, Dropdown Plugin, Scrollspy Plugin, Tab Plugin, Tooltip Plugin, Popover Plugin, Alert Plugin, Button Plugin, Collapse Plugin, Carousel Plugin, Affix Plugin.

3. **Java Script** - Introduction, Variable, Value, Data type, Operators and Expressions, Arrays, Decision making and Loops, Functions, Variable Scope and Objects, The Browser Environment – BOM & DOM, Events and Event Handling.

4. **JQUERY-** Introduction, Structure, Using jQuery and including .js file to HTML, Type of Selectors, Handling Events with jQuery, jQuery DOM Traversal, jQuery DOM Manipulation, jQuery Effects, APIs, Geo-location, Local Storage, Session Storage, History.

Course Outcome (CO): Upon completion of the course, the students should be able to

CO1: understand and describe the role of front-end development in modern web applications.

CO2: act like a professional front-end developer.

CO3: Able to analyze problems, seek for needed information, apply a solution, and verify it.

CO4: See the alternative ways for creating a front-end.

CO5: Able to discuss and take into use more front-end technologies.

Text Books:

1. Web Design with HTML, CSS, JavaScript and jQuery Set Paperback by Jon Duckett, 2014.
2. Mastering HTML, CSS &Javascript Web Publishing Paperback – 15 Jul 2016 by Laura Lemay, Rafe Colburn, Jennifer Kyrnin
3. Beginning HTML, XHTML, CSS, and JavaScript (Wrox Programmer to Programmer) Paperback – Import, 29 Dec 2009by Jon Duckett
4. HTML 5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, jQuery)

Mobile Computing (MCAE-404)

Credit: 3+0

Course Objective:

- To make the student understand the concept of mobile computing paradigm, its novel applications and limitations.
- To understand the typical mobile networking infrastructure through a popular GSM protocol.
- To understand the issues and solutions of various layers of mobile networks, namely MAC layer, Network Layer & Transport Layer.
- To understand the database issues in mobile environments & data delivery models.
- To understand the ad hoc networks and related concepts.
- To understand the platforms and protocols used in mobile environment.

Syllabus

1. **Introduction:** Issues in mobile computing, overview of wireless telephony: cellular concept, GSM: air-interface, channel structure, location management: HLR-VLR, hierarchical, handoffs, channel allocation in cellular systems, CDMA, GPRS.

2. **Wireless Networking, Wireless LAN Overview:** MAC issues, IEEE 802.11, Blue Tooth, Wireless multiple access protocols, TCP over wireless, Wireless applications, data broadcasting, Mobile IP, WAP: Architecture, protocol stack, application environment, applications.

3. **Data Management Issues & Mobile Agents:** Data replication for mobile computers, adaptive clustering for mobile wireless networks, File system, Disconnected operations. **Mobile Agents:** computing, security and fault tolerance, transaction processing in mobile computing environment.

4. **Ad Hoc Networks:** Localization, MAC issues, Routing protocols, global state routing (GSR), Destination sequenced distance vector routing (DSDV), Dynamic source routing (DSR), Ad Hoc on demand distance vector routing (AODV), Temporary ordered routing algorithm (TORA), QoS in Ad Hoc Networks, applications.

Course Outcome (CO): At the end of course, the student will be able to

CO1: Explain and discuss issues in mobile computing and illustrate overview of wireless telephony and channel allocation in cellular systems.

CO2: Explore the concept of Wireless Networking and Wireless LAN.

CO3: Analyze and comprehend Data management issues like data replication for mobile computers, adaptive clustering for mobile wireless networks and Disconnected operations.

CO4: Identify Mobile computing Agents and state the issues pertaining to security and fault tolerance in mobile computing environment.

CO5: Compare and contrast various routing protocols and will identify and interpret the performance of network systems using Adhoc networks.

Text books:

1. J. Schiller, Mobile Communications, Addison Wesley.
2. A. Mehrotra, GSM System Engineering.
3. M. V. D. Heijden, M. Taylor, Understanding WAP, Artech House.
4. Charles Perkins, Mobile IP, Addison Wesley.
5. Charles Perkins, Ad hoc Networks, Addison Wesley.

Introduction to Deep Learning (MCAE-405)

Credit: 3+0

Course Objective: Deep learning is a class of machine learning algorithms which enables computers to learn from examples. Deep learning techniques have been used successfully for variety of applications, including: automatic speech recognition, image recognition, natural language processing, drug discovery, and recommendation systems. In this course, students will learn the fundamentals of deep learning, and the main research activities in this field.

Syllabus

1. **Introduction to Deep Networks:** Introduction to machine learning- Linear models (SVMs and Perceptrons, logistic regression)- Intro to Neural Nets: What a shallow network computes- Training a network: loss functions, back propagation and stochastic gradient descent- Neural networks as universal function approximates. **DEEP NETWORKS:** History of Deep Learning- A Probabilistic Theory of Deep Learning- Backpropagation and regularization, batch normalization- VC Dimension and Neural Nets-Deep Vs Shallow Networks-Convolutional Networks- Generative Adversarial Networks (GAN), Semi-supervised Learning.

2. **Dimensionality Reduction:** Linear (PCA, LDA) and manifolds, metric learning – Auto encoders and dimensionality reduction in networks - Introduction to Convnet - Architectures – AlexNet, VGG, Inception, ResNet - Training a Convnet: weights initialization, batch normalization, hyper parameter optimization.

3. **Optimization and Generalization:** Optimization in deep learning– non-convex optimization for deep networks- Stochastic Optimization Generalization in neural networks- Spatial Transformer Networks- Recurrent networks, LSTM - Recurrent Neural Network Language Models- Word-Level RNNs & Deep Reinforcement Learning - Computational & Artificial Neuroscience.

4. **Case Study and Applications:** Image net- Detection-Audio Wave Net-Natural Language Processing Word2Vec - Joint Detection-Bioinformatics- Face Recognition- Scene Understanding-Gathering Image Captions.

Course Outcome (CO): At the end of course, the student will be able

CO1: To present the mathematical, statistical and computational challenges of building neural networks.

CO2: To study the concepts of deep learning.

CO3: To introduce dimensionality reduction techniques.

CO4: To enable the students to know deep learning techniques to support real-time applications.

CO5: To examine the case studies of deep learning techniques.

Text Books:

1. Cosma Rohilla Shalizi, Advanced Data Analysis from an Elementary Point of View, 2015.
2. Deng & Yu, Deep Learning: Methods and Applications, Now Publishers, 2013.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
4. Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

Computational Intelligence (MCAE-406)

Credit: 3+0

Course Objectives:

The course goal is to make students familiar with basic principles of various computational methods of data processing that can commonly be called computational intelligence (CI). Examples of CI are nature-inspired methods (neural nets, evolutionary algorithms), fuzzy systems, as well as various probabilistic methods under uncertainty (e.g. Bayesian models) and machine learning methods (e.g. reinforcement learning). After the course the students will be able to conceptually understand the important terms and algorithms of CI, such that they would be able to choose appropriate method(s) for a given task.

Syllabus

1. **Introduction:** Introduction to Artificial Intelligence-Search-Heuristic Search-A* algorithm-Game Playing- Alpha-Beta Pruning-Expert Systems-Inference-Rules-Forward Chaining and Backward Chaining- Genetic Algorithms.

2. **Knowledge Representation and Reasoning:** Proposition Logic - First Order Predicate Logic – Unification – Forward Chaining -Backward Chaining - Resolution – Knowledge Representation - Ontological Engineering - Categories and Objects – Events - Mental Events and Mental Objects - Reasoning Systems for Categories - Reasoning with Default Information - Prolog Programming. **UNCERTAINTY:** Non monotonic reasoning-Fuzzy Logic-Fuzzy rules-fuzzy inference-Temporal Logic-Temporal Reasoning-Neural Networks-Neuro-fuzzy Inference.

3. **Learning:** Probability basics - Bayes Rule and its Applications - Bayesian Networks – Exact and Approximate Inference in Bayesian Networks - Hidden Markov Models - Forms of Learning - Supervised Learning - Learning Decision Trees – Regression and Classification with Linear Models - Artificial NeuralNetworks – Nonparametric Models - Support Vector Machines - Statistical Learning - Learning with Complete Data - Learning with Hidden Variables- The EM Algorithm – Reinforcement Learning

4. **Intelligence and Applications:** Natural language processing-Morphological Analysis-Syntax Analysis-Semantic Analysis-All applications – Language Models - Information Retrieval – Information Extraction – Machine Translation – Machine Learning - Symbol-Based – Machine Learning: Connectionist – Machine Learning.

Course Outcome (CO): Upon completion of the course, the students will be able to

CO1: Provide a basic exposition to the goals and methods of Computational Intelligence.

CO2: Study of the design of intelligent computational techniques.

CO3: Apply the Intelligent techniques for problem solving.

CO4: Improve problem solving skills using the acquired knowledge in the areas of, reasoning, natural language understanding, computer vision, automatic programming and machine learning.

Text Books:

1. Stuart Russell, Peter Norvig, —Artificial Intelligence: A Modern Approach, Third Edition, Pearson Education / Prentice Hall of India, 2010.
2. Elaine Rich and Kevin Knight, —Artificial Intelligence, Third Edition, Tata McGraw-Hill, 2010.
3. Patrick H. Winston. "Artificial Intelligence", Third edition, Pearson Edition, 2006.
4. Dan W.Patterson, —Introduction to Artificial Intelligence and Expert Systems, PHI, 2006.
5. Nils J. Nilsson, —Artificial Intelligence: A new Synthesis, Harcourt Asia Pvt. Ltd., 2000.

Natural Language Processing (MCAE-407)

Credit: 3+0

Course Objective: Course Objectives: To understand the advanced concepts of Natural Language Processing and to be able to apply the various concepts of NLP in other application areas.

Syllabus

1. **Introduction:** Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance, WORD LEVEL ANALYSIS : Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

2. **Syntactic Analysis:** Context Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing – Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures.

3. **Semantics and Pragmatics:** Requirements for representation, First-Order Logic, Description Logics – Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods.

4. **Basic Concepts of Speech Processing:** Speech Fundamentals: Articulatory Phonetics – Production and Classification of Speech Sounds; Acoustic Phonetics – Acoustics of Speech Production; Review of Digital Signal Processing Concepts; Short-Time Fourier Transform, Filter-Bank and LPC Methods. **Speech-Analysis:** Features, Feature Extraction and Pattern Comparison Techniques: Speech Distortion Measures– Mathematical and Perceptual – Log–Spectral Distance, Cepstral Distances, Weighted Cepstral Distances and Filtering, Likelihood Distortions, Spectral Distortion Using a Warped Frequency Scale, LPC, PLP And MFCC Coefficients, Time Alignment and Normalization – Dynamic Time Warping, Multiple Time – Alignment Paths. SPEECH MODELING: Hidden Markov Models: Markov Processes, HMMs – Evaluation, Optimal State Sequence – Viterbi Search, Baum-Welch Parameter Re-Estimation, Implementation Issues.

Course Outcome (CO): At the end of course, the student will be able

CO1: To learn the fundamentals of natural language processing.

CO2: To understand the use of CFG and PCFG in NLP.

CO3: To understand the role of semantics of sentences and pragmatic.

CO4: To Introduce Speech Production and Related Parameters of Speech.

CO5: To show the computation and use of techniques such as short time Fourier transform, linear predictive coefficients and other coefficients in the analysis of speech.

Text Books:

1. Daniel Jurafsky, James H. Martin—Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014.
2. Steven Bird, Ewan Klein and Edward Loper, —Natural Language Processing with Python, First Edition, OReilly Media, 2009.
3. Lawrence Rabiner and Biing-Hwang Juang, "Fundamentals of Speech Recognition", Pearson

Education, 2003.
4. Daniel Jurafsky and James H Martin, "Speech and Language Processing – An Introduction To Natural Language Processing, Computational Linguistics, And Speech Recognition", Pearson Education, 2002.
5. Frederick Jelinek, "Statistical Methods of Speech Recognition", MIT Press, 1997.
6. Breck Baldwin, —Language Processing with Java and Ling Pipe Cookbook, Atlantic Publisher, 2015
7. Richard M Reese, —Natural Language Processing with Java, OReilly Media, 2015.
8. Nitin Indurkhya and Fred J. Damerau, —Handbook of Natural Language Processing, Second Edition, Chapman and Hall/CRC Press, 2010.
9. Tanveer Siddiqui, U.S. Tiwary, —Natural Language Processing and Information Retrieval, Oxford University Press, 2008.

Parallel and Distributed Computing (MCAE-408)

Credit: 3+0

Course Objective: To learn the advanced concepts of Parallel and Distributed Computing and its implementation for assessment of understanding the course by the students.

Syllabus

1. **Parallelism Fundamentals and Parallel Architecture:** Scope and issues of parallel and distributed computing, Parallelism, Goals of parallelism, Parallelism and concurrency, Multiple simultaneous computations, Programming Constructs for creating Parallelism, communication, and coordination. Programming errors not found in sequential programming like data races, higher level races, lack of liveness. **Parallel Architecture:** Architecture of Parallel Computer, Communication Costs, parallel computer structure, architectural classification schemes, Multicore processors, Memory Issues: Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, vector processing, GPU, coprocessing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).

2. **Parallel Decomposition and Parallel Performance:** Need for communication and coordination/synchronization, Scheduling and contention, Independence and partitioning, Task- Based Decomposition, Data Parallel Decomposition, Actors and Reactive Processes, Load balancing, Data Management, Impact of composing multiple concurrent components, Power usage and management. Sources of Overhead in Parallel Programs, Performance metrics for parallel algorithm implementations, Performance measurement, The Effect of Granularity on Performance Power Use and Management, Cost-Performance trade-off;

3. **Distributed Computing:** Definition, Relation to parallel systems, synchronous vs asynchronous execution, design issues and challenges, A Model of Distributed Computations, A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication. Communication and Coordination: Shared Memory, Consistency, Atomicity, Message- Passing, Consensus, Conditional Actions, Critical Paths, Scalability, cache coherence in multiprocessor systems, synchronization mechanism. **CUDA programming model:** Overview of CUDA, Isolating data to be used by parallelized code, API function to allocate memory on the parallel computing device, API function to transfer data to parallel computing device, Concepts of Threads, Blocks, Grids, developing kernel function that will be executed by threads in the parallelized part, Launching the execution of kernel function by parallel threads, transferring data back to host processor with API function call.

4. **Parallel Algorithms design, Analysis, and Programming:** Parallel Algorithms, Parallel Graph Algorithms, Parallel Matrix Computations, Critical paths, work and span and relation to Amdahl's law, Speed-up and scalability, naturally parallel algorithms, Parallel algorithmic patterns like divide and conquer, map and reduce, Specific algorithms like parallel Merge Sort, Parallel graph algorithms, parallel shortest path, parallel spanning tree, Producer-consumer and pipelined algorithms.

Course outcomes (CO): On completion of this course, the students will be able to

CO1: Apply the fundamentals of parallel and distributed computing including parallel architectures and paradigms.

CO2: Apply parallel algorithms and key technologies.

CO3: Develop and execute basic parallel and distributed applications using basic programming models and tools.

CO4: Analyze the performance issues in parallel computing and trade-offs.

Text Books:

1. C Lin, L Snyder. Principles of Parallel Programming. USA: Addison-Wesley (2008).
2. A Grama, A Gupra, G Karypis, V Kumar. Introduction to Parallel Computing, Addison Wesley (2003).
3. B Gaster, L Howes, D Kaeli, P Mistry, and D Schaa. Heterogeneous Computing with Opencl. Morgan Kaufmann and Elsevier (2011).
4. T Mattson, B Sanders, B Massingill. Patterns for Parallel Programming. Addison-Wesley (2004).
5. Quinn, M. J.,Parallel Programming in C with MPI and OpenMP, McGraw-Hill (2004)