BACHELOR OF TECHNOLOGY
CURRICULA & SYLLABI

Computer Science & Engineering Department

ABOUT THE DEPARTMENT

The department of Computer Science& Engineering was established in 1984. It offers a dynamic environment for study, research and professional growth. The present intake to B.Tech. course is 120. The department over the years has established its status as a centre for imparting High Quality education to B.Tech. students.

The department offers a three year Master of Computer Application (MCA) programme with an intake of 60 students. The department also offers 2 year regular M.Tech. courses in Computer Science & Engineering and Information Technology with intake of 18 in each besides the regular Ph.D. program. The department is also program study centre of Indira Gandhi National Open University (IGNOU) for conducting various Computer courses such as MCA & BCA.

VISION

To become a leader of education, research and innovation in the area of Computer Science and Engineering and to produce under graduates who are globally recognized as innovative and well prepared computing professionals.

MISSION

1. To create, share and disseminate knowledge through research and education in the theory and application of computing.
2. To train the students in different aspects of computing discipline for enhancing, augmenting and updating their technical skills
3. To inculcate the spirit of analysis, team work, innovation and professionalism among the students.

PROGRAMME EDUCATIONAL OBJECTIVES (PEO) OF B.TECH. PROGRAMME

PEO-1 To inculcate the knowledge of the fundamentals of the mathematics, science & engineering disciplines for developing the ability to formulate, solve and analyze

the problems of Computer Science & Engineering field and to provide them the skills for the pursuit of under-graduate studies, research and development and higher education.

PEO-2 To provide the understanding of the prerequisite of the software, technical aspects and design for coming up with the novel engineering solutions and efficient product developments.

PEO-3 To assist the students in the pursuit of the successful career by adopting the ethical practices and social responsibility.

PEO-4 To provide students the technical as well as soft skills required by the national as well as international organizations.

PEO-5 To elevate cognizance in the students toward the unending learning and to inculcate the ethical and moral ways.

PEO-6 To give students the knowledge of the contemporary technologies, practical experiences and possibilities in the field of Computer Science & Engineering and to provide the multidisciplinary knowledge to develop the team spirit and leadership qualities by working on multidisciplinary projects.

PROGRAM OUTCOMES (POs) of B.Tech. PROGRAMME:

PO-1 The students will develop the ability towards the application of fundamental knowledge of computing, mathematics, algorithms and computer science & engineering precepts and rationales for developing the solutions of the critical engineering problems. (Rudimentary engineering analytical skills).

PO-2 The under-graduating students will be able to model and carry out the experiments by using the fundamental knowledge of computer science & engineering discipline and derive the conclusions by analysing and interpreting the data.

PO-3 The students will be able to analyze, design, implement and assess a computer based information system, procedure, module or program to fulfil the requirements along with the consideration of economical, social, privacy and reliability constraints.(innovative skills)

PO-4 The students will be able to perform efficaciously in multi-disciplinary teams. (Team spirit)

PO-5 The students will develop the analytical skills to critically analyze, recognize, formulate and devise solutions to the engineering problems by using the adequate computing and engineering skills and knowledge. (Engineering problem solving skills)

PO-6 The students will have the awareness towards the professional, ethical practices, legal, security & social consequences and obligation. (Professional integrity).

PO-7 The students will have the efficient speaking skill and written/interpersonal communication skills.(Oral & written communication skill)

PO-8 To impart the exhaustive education in the students required to understand and analyze the local and global consequences of computer science & engineering solutions ranging from individuals and organizations to society. (Engineering consequences assessment skills)

PO-9 The students will develop the realization of the requirement of and the ability to indulge in maintaining professional growth and unending learning. (Continuing education cognizance).

PO-10 The students will have the cognition towards the current issues and problems. (Societal awareness)

PO-11 The students will possess the ability to utilize the knowledge of innovative computing equipments required for engineering tasks. (Pragmatic skills)

PO-12 The students will be able to apply the design and evolution precepts in the development of software and hardware computer systems of variable complications. (Software hardware interface).

Credit Structure for B. Tech. (Computer Science & Engineering)

(For newly admitted students from Session 2014-2015)

Category / Semesters	I	II	III	IV	V	VI	VII	VIII	Total
Basic Sciences & Maths (BSM)	9	14	9	4	-	-	-	-	**36**
Engineering Fundamentals (EF)	12	7	6	2	-	-	-	-	**27**
Department Core (DC)	-	-	10	14	20	25	10	5	**84**
Management (M)	-	-	-	3	3	-	-	-	**6**
Humanities & Social Science Core (HSSC)	4	-	-	-	-	-	-	-	**4**
Project (P)	-	-	-	-	-	-	5	5	**10**
Programme Electives (PE)	-	-	-	-	-	-	8	8	**16**
Open Electives (OE)	-	-	-	-	-	-	-	4	**4**
Humanities & Social Science Electives (HSSE)	-	3	-	-	-	-	-	-	**3**
Total	**25**	**24**	**25**	**23**	**23**	**25**	**23**	**22**	**190**

Curriculum for B.Tech. (Computer Science & Engineering)

Freshman Year, Semester-I

S.N.	Category	Paper Code	Subject	L	T	P	Credits
1.	BSM	BAS-01	Engineering Mathematics-I	3	1	0	4
2.	BSM	BAS-02	Engineering Physics-I	3	1	2	5
3.	EF	BCS-02	Introduction to C & Functional Programming	3	1	2	5
4.	EF	BEE-01	Principles of Electrical Engineering	3	1	2	5
5.	HSSC	BAS-03	Professional Communication	3	1	0	4
6.	EF	BCS-03	Software Lab-I	0	0	4	2
7.	AC	BAS-05	Environment & Ecology	2	1	0	-
			Total	**15**	**5**	**10**	**25**

Freshman Year, Semester-II

S.N.	Category	Paper Code	Subject	L	T	P	Credits
1.	BSM	BAS-07	Engineering Mathematics-II	3	1	0	4
2.	BSM	BAS-08	Engineering Physics-II	3	1	2	5
3.	BSM	BAS-14	Graph Theory	3	1	2	5
4.	EF	BCS-04	Object Oriented Modeling & C++	3	1	2	5

5.	HSSE	BAS-**	Humanities & Social Science Electives	2	1	0	3
6.	EF	BCS-05	Software Lab-II	0	0	4	2
7.	AC	BEC-01	Fundamentals of Electronics Engineering	3	1	2	-
			Total	**14**	**5**	**10**	**24**

Sophomore Year, Semester-III

S.N.	Category	Paper Code	Subject	L	T	P	Credits
1.	BSM	BAS-27	Discrete Mathematics	3	1	0	4
2.	BSM	BAS-24	Applied Computational Methods	3	1	2	5
3.	EF	BCS-11	Digital Circuits and Logic Design	3	1	0	4
4.	DC	BCS-12	Principles of Data Structures through C/C++	3	1	2	5
5.	DC	BCS-13	Internet & JAVA Programming	3	1	2	5
6.	EF	BCS-14	Software Lab-III	0	0	4	2
7.	AC	MAS 109/110/ 111	One of the Foreign Languages (French, German, Spanish etc.)	3	1	0	-
			Total	**15**	**5**	**10**	**25**

Sophomore Year, Semester-IV

S.N.	Category	Paper Code	Subject	L	T	P	Credits
1.	BSM	BAS-26	Optimization Techniques	3	1	0	4
2.	M	MBA-113	Management Information System	2	1	0	3
3.	DC	BCS-15	Database Management Systems	3	1	2	5
4.	DC	BCS-16	Theory of Computation	3	1	0	4
5.	DC	BCS-17	Computer Organization & Design	3	1	2	5
6.	EF	BCS-18	Software Lab-IV	0	0	4	2
7.	AC	BEC-32	Microprocessors & Application	3	1	2	-
			Total	**14**	**5**	**8**	**23**

Junior Year, Semester-V

S.N.	Category	Paper Code	Subject	L	T	P	Credits
1.	M	MBA-02	Engineering and Managerial Economics	2	1	0	3
2.	DC	BCS-26	Principles of Operating Systems	3	1	2	5
3.	DC	BCS-27	Computer Graphics	3	1	2	5
4.	DC	BCS-28	Design & Analysis of Algorithms	3	1	2	5
5.	DC	BCS-29	Advanced Computer Architecture	3	1	2	5
6.	AC	BEC-42	Digital Signal Processing	3	1	0	-
			Total	**14**	**5**	**8**	**23**

Junior Year, Semester-VI

S.N.	Category	Paper Code	Subject	L	T	P	Credits
1.	DC	BCS-31	Principle of Compiler Design	3	1	2	5
2.	DC	BCS-32	Artificial Intelligence	3	1	2	5
3.	DC	BCS-33	Web Technologies	3	1	2	5
4.	DC	BCS-34	Computer Networks	3	1	2	5
5.	DC	BCS-35	Software Engineering	3	1	2	5
6.	AC	BCS-30	Seminar	0	0	6	-
			Total	**15**	**5**	**10**	**25**

Senior Year, Semester-VII

S.N.	Category	Paper Code	Subject	L	T	P	Credits
1.	DC	BCS-41	Introduction to Machine Learning	3	1	2	5
2.	DC	BCS-42	Parallel & Distributed Computing	3	1	2	5
3.	PE1	BCS-**	Programme Elective-1	3	1	0	4
4.	PE2	BCS-**	Programme Elective-2	3	1	0	4
5.	P	BCS-40	Project Part-I	0	0	10	5
6.	AC	BCS-45	Industrial/Practical Training	0	0	2	-
			Total	**12**	**4**	**14**	**23**

Senior Year, Semester-VIII

S.N.	Category	Paper Code	Subject	L	T	P	Credits
1.	DC	BCS-43	Mobile Computing	3	1	2	5
2.	PE3	BCS-**	Programme Elective-3	3	1	0	4
3.	PE4	BCS-**	Programme Elective-4	3	1	0	4
4.	OE	BOE-**	Open Elective offered by other Department	3	1	0	4
5.	P	BCS-50	Project Part-II	0	0	10	5
			Total	**12**	**4**	**12**	**22**

Engineering Fundamentals & Department Core (Computer Science & Engineering)

S.N.	Paper Code	Subject	Prerequisite Subject	L	T	P	Credits
		I Year					

1.	BCS-02	Introduction to C & Functional Programming	-	3	1	2	5
2.	BCS-03	Software Lab-I	-	0	0	4	2
3.	BCS-04	Object Oriented Modeling & C++	-	3	1	2	5
4.	BCS-05	Software Lab-II	-	0	0	4	2
		II Year					
5.	BCS-11	Digital Circuits and Logic Design	-	3	1	0	4
6.	BCS-12	Principles of Data Structures through C/C++	-	3	1	2	5
7.	BCS-13	Internet & JAVA Programming	-	3	1	2	5
8.	BCS-14	Software Lab-III	-	0	0	4	2
9.	BCS-15	Database Management Systems	-	3	1	2	5
10.	BCS-16	Theory of Computation	-	3	1	0	4
11.	BCS-17	Computer Organization & Design	-	3	1	2	5
12.	BCS-18	Software Lab-IV	-	0	0	4	2
		III Year					
13.	BCS-26	Principles of Operating Systems	-	3	1	2	5
14.	BCS-27	Computer Graphics	-	3	1	2	5
15.	BCS-28	Design & Analysis of Algorithms	-	3	1	2	5
16.	BCS-29	Advanced Computer Architecture	-	3	1	2	5
17.	BCS-31	Principle of Compiler Design	-	3	1	2	5
18.	BCS-32	Artificial Intelligence	-	3	1	2	5
19.	BCS-33	Web Technologies	-	3	1	2	5
20.	BCS-34	Computer Networks	-	3	1	2	5
21.	BCS-35	Software Engineering	-	3	1	2	5
22.	BCS-30	Seminar	-	0	0	6	-
		IV Year					
23.	BCS-41	Introduction to Machine Learning	-	3	1	2	5
24.	BCS-42	Parallel & Distributed Computing	-	3	1	2	5
25.	BCS-40	Project Part-I	-	0	0	10	5
26.	BCS-45	Industrial/Practical Training	-	0	0	2	-
27.	BCS-43	Mobile Computing	-	3	1	2	5
28.	BCS-50	Project Part-II	BCS-40	0	0	10	5

Programme Electives (Computer Science & Engineering)

S.N.	Paper Code	Subject	Prerequisite Subject	L	T	P	Credits
		PE1 & PE2 (VII Semester)					
1.	BCS-51	Advanced JAVA	BCS-13	3	1	0	4

2.	BCS-52	.Net Technology	-	3	1	0	4
3.	BCS-53	LAMP Technology	-	3	1	0	4
4.	BCS-54	Network Programming	BCS-34	3	1	0	4
5.	BCS-55	Mobile Application Programming	BCS-34	3	1	0	4
6.	BCS-56	Linux Administration & System Call Programming	BCS-26	3	1	0	4
7.	BCS-57	Database Administration with ORACLE	BCS-15	3	1	0	4
8.	BCS-58	Data warehousing & Data Mining	BCS-15	3	1	0	4
9.	BCS-59	Analytics and Systems of Big Data	BCS-15	3	1	0	4
10.	BCS-60	Game Theory	-	3	1	0	4
		PE3 & PE4 (VIII Semester)					
11.	BCS-66	Advanced Programming Techniques	-	3	1	0	4
12.	BCS-67	Computer Vision: Foundations and Applications	-	3	1	0	4
13.	BCS-68	Software Reuse	BCS-35	3	1	0	4
14.	BCS-69	Software Verification & Validation	BCS-35	3	1	0	4
15.	BCS-70	Software Design & Construction	BCS-35	3	1	0	4
16.	BCS-71	Software Quality Management	BCS-35	3	1	0	4
17.	BCS-72	Aspect Oriented Programming	-	3	1	0	4
18.	BCS-73	Neural Networks & Fuzzy Systems	-	3	1	0	4
19.	BCS-74	Fundamentals of Cloud Computing	BCS-26 BCS-34	3	1	0	4
20.	BCS-75	Advanced Multi-core Systems	BCS-28	3	1	0	4
21.	BCS-76	Cryptography & Information Security	-	3	1	0	4
22.	BCS-77	Digital Image Processing	BEC-42	3	1	0	4
23.	BCS-78	High Performance Computing	-	3	1	0	4
24.	BCS-79	Introduction to Real Time Systems	-	3	1	0	4

Subjects for other Departments

S.N.	Paper Code	Subject	Prerequisite Subject	L	T	P	Credits
1.	BCS-01	Introduction to Computer Programming	-	3	1	2	5
2.	BCS-19	Web Designing	-	0	0	4	2
3.	BCS-36	Database Management System, Data Mining & Warehousing	-	3	1	0	4
4.	BCS-37	Network Security & Cryptography	-	3	1	2	5
5.	BCS-44	Object Oriented Techniques & JAVA Programming	-	3	1	0	4
6.	BOE-07	Introduction to Data & File Structures	-	2	1	2	4
7.	BOE-08	Introduction to Web Technology	-	2	1	2	4
8.	BOE-09	Linux & Shell Programming	-	2	1	2	4

Humanities & Social Science Electives

S.N.	Paper Code	Subject	Prerequisite Subject	L	T	P	Credits
1.	BAS-11	Human Values & Professional Ethics	-	2	1	0	3
2.	BAS-12	Industrial Psychology	-	2	1	0	3
3.	BAS-13	Industrial Sociology	-	2	1	0	3

Computer Science &Engineering Department

Subjects Offered by the Department

S.N.	Paper Code	Subject	Prerequisite Subject	L	T	P	Credits
1.	BCS-01	Introduction to Computer Programming	-	3	1	2	5
2.	BCS-02	Introduction to C & Functional Programming	-	3	1	2	5
3.	BCS-03	Software Lab-I	-	0	0	4	2
4.	BCS-04	Object Oriented Modeling & C++	-	3	1	2	5
5.	BCS-05	Software Lab-II	-	0	0	4	2
6.	BCS-11	Digital Circuits and Logic Design	-	3	1	0	4
7.	BCS-12	Principles of Data Structures through C/C++	-	3	1	2	5
8.	BCS-13	Internet & JAVA Programming	-	3	1	2	5
9.	BCS-14	Software Lab-III	-	0	0	4	2
10.	BCS-15	Database Management Systems	-	3	1	2	5
11.	BCS-16	Theory of Computation	-	3	1	0	4
12.	BCS-17	Computer Organization & Design	-	3	1	2	5
13.	BCS-18	Software Lab-IV	-	0	0	4	2
14.	BCS-19	Web Designing	-	0	0	4	2
15.	BCS-26	Principles of Operating Systems	-	3	1	2	5
16.	BCS-27	Computer Graphics	-	3	1	2	5
17.	BCS-28	Design & Analysis of Algorithms	-	3	1	2	5
18.	BCS-29	Advanced Computer Architecture	-	3	1	2	5
19.	BCS-30	Seminar	-	0	0	6	-
20.	BCS-31	Principle of Compiler Design	-	3	1	2	5
21.	BCS-32	Artificial Intelligence	-	3	1	2	5
22.	BCS-33	Web Technologies	-	3	1	2	5
23.	BCS-34	Computer Networks	-	3	1	2	5

24.	BCS-35	Software Engineering	-	3	1	2	5
25.	BCS-36	Database Management System, Data Mining & Warehousing	-	3	1	0	4
26.	BCS-37	Network Security & Cryptography	-	3	1	2	5
27.	BCS-40	Project Part-I	-	0	0	10	5
28.	BCS-41	Introduction to Machine Learning	-	3	1	2	5
29.	BCS-42	Parallel & Distributed Computing	-	3	1	2	5
30.	BCS-43	Mobile Computing	-	3	1	2	5
31.	BCS-44	Object Oriented Techniques & JAVA Programming	-	3	1	0	4
32.	BCS-45	Industrial/Practical Training	-	0	0	2	-
33.	BCS-50	Project Part-II	BCS-40	0	0	10	5
33.	BCS-51	Advanced JAVA	BCS-13	3	1	0	4
34.	BCS-52	.Net Technology	-	3	1	0	4
35.	BCS-53	LAMP Technology	-	3	1	0	4
36.	BCS-54	Network Programming	BCS-34	3	1	0	4
37.	BCS-55	Mobile Application Programming	BCS-34	3	1	0	4
38.	BCS-56	Linux Administration & System Call Programming	BCS-26	3	1	0	4
39.	BCS-57	Database Administration with ORACLE	BCS-15	3	1	0	4
40.	BCS-58	Data Warehousing & Data Mining	BCS-15	3	1	0	4
41.	BCS-59	Analytics and Systems of Big Data	BCS-15	3	1	0	4
42.	BCS-60	Game Theory	-	3	1	0	4
43.	BCS-66	Advanced Programming Techniques	-	3	1	0	4
44.	BCS-67	Computer Vision: Foundations and Applications	-	3	1	0	4
45.	BCS-68	Software Reuse	BCS-35	3	1	0	4
46.	BCS-69	Software Verification & Validation	BCS-35	3	1	0	4
47.	BCS-70	Software Design & Construction	BCS-35	3	1	0	4
48.	BCS-71	Software Quality Management	BCS-35	3	1	0	4
49.	BCS-72	Aspect Oriented Programming	-	3	1	0	4
50.	BCS-73	Neural Networks & Fuzzy Systems	-	3	1	0	4
51.	BCS-74	Fundamentals of Cloud Computing	BCS-26 BCS-34	3	1	0	4
52.	BCS-75	Advanced Multi-core Systems	BCS-28	3	1	0	4
53.	BCS-76	Cryptography & Information Security	-	3	1	0	4
54.	BCS-77	Digital Image Processing	BEC-42	3	1	0	4
55.	BCS-78	High Performance Computing	-	3	1	0	4
56.	BCS-79	Introduction to Real Time Systems	-	3	1	0	4

SYLLABI

BCS-01 INTRODUCTION TO C PROGRAMMING

Course Category	:	Engineering Fundamental (EF) for other Departments
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Read and understand C programs.
2. Discuss basic theory and practice of programming.
3. Design and implement practical programs using C language.
4. Use compiler and feel comfortable with Windows environment
5. Identify and fix common C errors

Topics Covered

UNIT-I

Basics of Computer: Introduction to Digital Computer, Basic Operations of Computer, Functional Components of Computer, Classification of Computers. Introduction to Operating System: DOS, Windows, Linux, Function, Services and Types. Basics of Programming: Approaches to Problem Solving, Concept of Algorithm and Flow Charts, Types of Computer Languages:- Machine Language, Assembly Language and High Level Language, Concept of Assembler, Compiler, Loader and Linker. 9

UNIT-II

Standard I/O in "C", Fundamental Data Types and Storage Classes: Character Types, Integer, Short, Long, Unsigned, Single and Double-Precision Floating Point, Storage Classes, Automatic, Register, Static and External, Operators and Expressions: Using Numeric and Relational Operators, Mixed Operands and Type Conversion, Logical Operators, Bit Operations, 9

Operator Precedence and Associativity, C Conditional Program Execution: Applying if and Switch Statements, Nesting if and else, Restrictions on switch Values, Use of Break, Program Loops and Iteration: Uses of while, do and for Loops, Multiple Loop Variables, Assignment Operators, Using Break and Continue

UNIT-III

Arrays: One Dimensional, Multidimensional Array and their Applications, Declaration and Manipulation of Arrays Structures: Purpose and Usage of Structures, Declaring Structures, Assigning of Structures, Strings: String Variable, String Handling Functions, Array of Strings, Functions: Designing Structured Programs, Functions in C, User Defined and Standard Functions, Formal vs. Actual Arguments, Function Category, Function Prototype, Parameter Passing, Recursive Functions. Storage Classes: Auto, Extern, Register and Static Variables 9

UNIT-IV

Pointers: Pointer Variable and its Importance, Pointer Arithmetic and Scale Factor, Compatibility, Dereferencing, L value and R-Value, Pointers and Arrays, Pointer and Character Strings, Pointers and Functions, Array of Pointers, Pointers to Pointers Dynamic Memory Allocation Structure and Union: Declaration and Initialization of Structures, Structure as Function Parameters, Structure Pointers, Unions. File Management: Defining and Opening A File, Closing A File, Input/Output Operations in Files, Pre-Processor Directives, Command Line Arguments. 9

EXPERIMENTS

1. Write a program that finds whether a given number is even or odd.
2. Write a program that tells whether a given year is a leap year or not.
3. Write a program that accepts marks of five subjects and finds percentage and prints grades according to the following criteria:
    a. Between 90-100%----------Print ‘A’
    b. 80-90%----------Print ‘B’
    c. 60-80%----------Print ‘C’
    d. Below 60%----------Print ‘D’
4. Write a program that takes two operands and one operator from the user and perform the operation and prints the result by using Switch statement.
5. Write a program to print sum of even and odd numbers from 1 to N numbers.
6. Write a program to print the Fibonacci series.
7. Write a program to check whether the entered number is prime or not.
8. Write a program to find the reverse of a number.
9. Write a program to print Armstrong Numbers from 1 to 100.
10. Write a program to convert binary number into decimal number and vice versa.
11. Write a program that simply takes elements of the array from the user and finds the sum of these elements.
12. Write a program that inputs two arrays and saves sum of corresponding elements of these arrays in a third array and prints them.
13. Write a program to find the minimum and maximum element of the array.
14. Write a program to search an element in array using Linear Search.
15. Write a program to sort the elements of the array in ascending order using Bubble Sort technique.

16. Write a program to add and multiply two matrices of order NxN.
17. Write a program that finds the sum of diagonal elements of a MxN matrix.
18. Define a structure data type TRAIN_INFO. The type contain
    a. Train No.: integer type
    b. Train name: string
    c. Departure Time: aggregate type TIME
    d. Arrival Time : aggregate type TIME
    e. Start station: string
    f. End station : string

    The structure type Time contains two integer members: hour and minute. Maintain a train Time table and
19. implement the following operations:
    i. List all the trains (sorted according to train number) that depart from a particular section.
    ii. List all the trains that depart from a particular station at a particular time.
    iii. List all he trains that depart from a particular station within the next one hour of a given time.
    iv. List all the trains between a pair of start station and end station.
20. Write a program to swap two elements using the concept of pointers.
21. Write a program to compare the contents of two files and determine whether they are same or not.
22. Write a program to check whether a given word exists in a file or not. If yes then find the number of times it occurs.

Textbooks

1. Jeri R. Hanly, Elliot B. Koffman, Problem Solving and Program Design in C, 7th edition, Pearson
2. Childt , Herbert Complete reference with C Tata McGraw Hill

Reference books

1. Kerninghan and Ritchie, The C programming language, Prentice Hall
2. Samuel P. Harbison, and Guy L. Steele Jr., C-A Reference Manual, Fifth Edition, Prentice Hall, 2002

BCS-02 INTRODUCTION TO C & FUNCTIONAL PROGRAMMING

Course Category	:	Engineering Fundamental (EF)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1, Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the

following knowledge, skills and attitudes after completing this course.

1. Basic Terminology used in Computer Programming.
2. Programs Development in C Language by Writing, Compiling and Debugging.
3. Design of Programs involving Simple Statements, Conditional Statements, Iterative Statements, Array, Strings, Functions, Recursion, Structure and Union.
4. Difference between Call by Value and Call by Reference.
5. Dynamic Memory Allocations and Use of Pointers.
6. Basic Operations on a File.
7. Basics of Functional Programming.

Topics Covered

UNIT-I

Basics of Programming: Approaches to Problem Solving, Concept of Algorithm and Flow Charts, Types of Computer Languages-Machine Language, Assembly Language and High Level Language, Concept of Assembler, Compiler, Loader and Linker. 9

Data types, Operators, Expressions, Operator Precedence and Associativity

Fundamentals of C Programming: Structure of C Program, Writing and Executing the First C Program, Components of C Language. Standard I/O in C

Conditional program execution: Applying if and switch Statements, Nesting if and else

Program Loops and Iterations: use of while, do while and for Loops, Multiple Loop Variables, Use of break and continue Statements.

UNIT-II

Arrays: One Dimensional, Multidimensional Array and Their Applications, Declaration and Manipulation of Arrays 9

Strings: String Variable, String Handling Functions, Array of Strings

Functions: Designing Structured Programs, Functions in C, User Defined and Standard Functions, Formal vs. Actual Arguments, Function Category, Function Prototype, Parameter Passing, Recursive Functions.

Storage classes: Auto, Extern, Register and Static Variables

UNIT-III

Pointers: Pointer Variable and its Importance, Pointer Arithmetic Pointers and Arrays, Pointer and Character Strings, Pointers and Functions, Array of Pointers, Pointers to Pointers, Dynamic Memory Allocation 9

Structure and Union: Declaration and Initialization of Structures, Structure as Function Parameters, Structure Pointers, Unions.

File Management: Defining and Opening a File, Closing a File, Input/ Output Operations in Files, Random Access to Files, Error Handling

The Pre-processor Directives, Command Line Arguments, Macros

UNIT-IV

Principles of Functional Programming: Expressions, Evaluations, Functions and Types 9

Type Definitions and Built-in Types: Numbers, Characters, Strings and Lists. Basic Operations on Lists, Including Map, Fold And Filter, together with Their Algebraic Properties. Recursive Definitions and Structural Induction, Simple Program Calculation, Infinite Lists and Their Uses

EXPERIMENTS

1. Write a program to find the nature of the roots as well as value of the roots. However, in case of imaginary roots, find the real part and imaginary part separately.
2. Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a program to generate the first n terms of the sequence. For example, for n = 8, the output should be 0 1 1 2 3 5 8 13
3. Write a program to print all the prime numbers between m and n, where the value of m and n is supplied by the user.
4. The number such as 1991 is a palindrome because it is same number when read forward or backward. Write a program to check whether the given number is palindrome or not.
5. A positive integer number IJK is said to be well-ordered if I < J < K. For example, number 138 is called well-ordered because the digits in the number (1, 3, 8) increase from left to right, i.e., 1 < 3 < 8. Number 365 is not well-ordered because 6 is larger than 5. Write a program that will find and display all possible three digit well-ordered numbers. The program should also display the total number of three digit well-ordered numbers found.
6. Write a function to computer the highest common factor of integer numbers m and n. Use this function to find the highest common factor of integer numbers a and b.
7. Write a program to multiply matrix A (m×n) by B (p×q), given that n = p.
8. Write a program to sort a list of n integer numbers in descending order using bubble sort method.

Textbooks

1. Jeri R. Hanly and Elliot B. Koffman, Problem Solving and Program Design in C, 7th Edition, Pearson
2. Schildt , Herbert, Complete Reference with C, Tata McGraw Hill
3. Kerninghan and Ritchie,The C programming Language, 2nd Edition, Prentice Hall
4. Richard Bird, Introduction to Functional Programming using Haskell, 2nd Edition, Prentice-Hall International, 1998

Reference books

1. Greg Michaelson, An Introduction to Functional Programming Through Lambda Calculus, Dover Edition, Addition Wesley Publication
2. Samuel P. Harbison, and Guy L. Steele Jr., C-A Reference Manual, Fifth Edition, Prentice Hall, 2002

BCS-03 SOFTWARE LAB-I

Course Category : Engineering Fundamental (EF)
Pre-requisite Subject : NIL
Contact Hours/Week : Lecture : 0, Tutorial : 0, Practical: 4

Number of Credits	:	2
Course Assessment Methods	:	Continuous assessment through three Viva voce, Practical work/record, attendance and Major Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Basic Terminology used in C, Use of Standard C Library and Evaluation of Expressions.
2. Programming in C Language by Writing, Compiling and Debugging.
3. Designing of Programs involving Simple Statements, Conditional Statements, Iterative Statements, Array, Strings, Functions, Recursion, Structure and Union.
4. Basic Pointer Programming.
5. Programming for Searching and Sorting.
6. Basic Operations on a File.
7. Basic Knowledge of Functional Programming.

EXPERIMENTS

The students should write programs in C to get the familiarization with following topics.

1. Get Familiar with C Compiler
2. Implement and Test Small Routine in C
3. Evaluation of Expression
4. Iteration, Function and Recursive Function
5. Arrays
6. Structures and Union
7. Searching and Selection
8. Sorting,
9. Strings Handling
10. Basic Pointer Programming
11. Files
12. Use of Standard C Library
13. Basics of Functional Programming.

BCS-04 OBJECT ORIENTED MODELING & C++

Course Category	:	Engineering Fundamental (EF)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment	:	Continuous assessment through tutorials, attendance, home

Methods		assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Understand the Concept of Object Oriented Programming and Master OOP using C++.
2. Implementing the Real Life Problems using Object Oriented Techniques.
3. Improvement in Problem Solving Skills.

Topics Covered

UNIT-I

Object Oriented Programming Concepts – Objects, Classes, Methods and Messages- Abstraction and Encapsulation, Inheritance- Abstract Classes, Polymorphism. Introduction to C++- Objects-Classes- Constructors and Destructors. 9

UNIT-II

Operator Overloading - Friend Functions- Type Conversions- Templates - Inheritance – Virtual Functions- Runtime Polymorphism. Exception Handling - Streams and Formatted I/O 9

UNIT-III

Object Modeling: Objects and Classes, Links and Associations, Generalization and Inheritance, Aggregation, Abstracts Class, Multiple Inheritance, Meta Data, Candidate Keys, Constraints. Dynamics Modeling: Events and States, Operations, Nested State Diagrams and Concurrency. 9

UNIT-IV

Functional Modeling: Data Flow Diagram, Specifying Operations, Constraints, A Sample Functional Model. OMT (Object Modeling Techniques) Methodologies, Examples and Case Studies to Demonstrate Methodologies, Comparisons of Methodologies, SA/SD, JSD. 9

EXPERIMENTS

Write C++ Programs to illustrate the concept of the following:

1. Arrays
2. Structures
3. Pointers
4. Objects and Classes
5. Console I/O Operations
6. Scope Resolution and Memory Management Operators
7. Inheritance
8. Polymorphism
9. Virtual Functions
10. Friend Functions
11. Operator Overloading
12. Function Overloading
13. Constructors and Destructors
14. this Pointer
15. File I/O Operations

Analyze, Design and Develop Code for the Following System (one for a batch of three students) using Object Oriented Methodology

1. ATM (Automated Teller Machine) System
2. Online Reservation System
3. Online Quiz System
4. Stock Maintenance System
5. Course Registration System
6. Payroll System
7. Expert System
8. Library Management System
9. Real Time Scheduler
10. Online Purchase System

Textbooks

1. B. Trivedi Programming with ANSI C++, Oxford University Press, 2007.
2. Ira Pohl , Object Oriented Programming using C++, Pearson Education, Second Edition
3. B. Stroustrup, The C++ Programming Language, 3rdedition, Pearson Education, 2004
4. James Rumbaugh, et. al Object Oriented Modeling and Design-, PHI
5. Robert Lafore, Object Oriented Programming in Turbo C++, Galgotia Publication, 1994
6. E. Balaguruswamy, Object Oriented Programming with C++, TMH Publication
7. Grady Booch, James Rumbaugh and Ivar Jacobson The Unified Modeling Language User Guide, Pearson Education
8. Booch, Maksimchuk, Engle, Young, Conallen and Houstan, Object Oriented Analysis and Design with Applications, Pearson Education
9. S. B. Lippman, Josee Lajoie, Barbara E. Moo,C++ Primer,4th edition, Pearson Education, 2005

Reference books

1. Coleman, D. et.al. Object-Oriented Development, The Fusion Method. Prentice Hall

2. Booch, G. Object-Oriented Design with Applications. Redwood City, Bengamin/Cummings
3. Sartaj Sahni, Data Structures, Algorithms and Applications in C++, McGraw Hill, Second Edition, 2005.

BCS-05 SOFTWARE LAB-II

Course Category	:	Engineering Fundamental (EF)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 0, Tutorial : 0 , Practical: 4
Number of Credits	:	2
Course Assessment Methods	:	Continuous assessment through three Viva voce, Practical work/record, attendance and Major Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Differentiate between structures oriented programming and object oriented programming.
2. Use object oriented programming language like C++ and associated libraries to develop object oriented programs
3. Understand and apply various object oriented features like inheritance, data abstraction, encapsulation and polymorphism to solve various computing problems using C++ language.
4. Apply concepts of operator-overloading, constructors and destructors
5. Reuse the code and write the classes which work like built-in types.
6. Apply object -oriented concepts in real world applications.

EXPERIMENTS

To write following programs in C++

1. Using basic statements like control statements, looping statements, various I/O statements and various data structures.
2. Creating classes in C++ for understanding of basic OOPS features.
3. Representing concepts of data hiding, function overloading and operator overloading.
4. Using memory management features and various constructors and destructors.
5. Representing Inheritance, virtual classes and polymorphism.
6. Writing generic functions.
7. File handling programs.
8. Design and Implementation of some real life problems using Object Oriented Techniques (Object Model/Dynamic Model/Functional Model).

BCS-11 DIGITAL CIRCUITS AND LOGIC DESIGN

Course Category	:	Engineering Fundamental (EF)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Design a finite state machine and sequential logic design.
2. Synthesize a logic design from a natural language description of a problem.
3. Realize a complete arithmetic and logic unit.
4. Generate a realization of combinational logic in a programmable gate array.
5. Simulate a complete design to evaluate functional correctness and timing.

Topics Covered

UNIT-I

Binary Codes - Weighted and Non-Weighted - Binary Arithmetic Conversion Algorithms - Error Detecting and Error Correcting Codes - Canonical and Standard Boolean Expressions - Truth Tables. 9

UNIT-II

K-Map Reduction - Don't Care Conditions - Adders / Subtractors- Carry Look-Ahead Adder - Code Conversion Algorithms - Design of Code Converters - Equivalence Functions. 9

Binary/Decimal Parallel Adder/Subtractor for Signed Numbers - Magnitude Comparator - Decoders / Encoders - Multiplexers / Demultiplexers- Boolean Function Implementation using Multiplexers.

UNIT-III

Sequential Logic - Basic Latch - Flip-Flops (SR, D, JK, T and Master-Slave) - Triggering of Flip-Flops - Counters - Design Procedure - Ripple Counters - BCD and Binary - Synchronous Counters. 9

UNIT-IV

Registers - Shift Registers - Registers with Parallel Load - Memory Unit - Examples of RAM, ROM, PROM, EPROM - Reduction of State and Flow Tables - Race-Free State Assignment - Hazards. 9

Textbooks

1. Morris Mano, Digital Design, Prentice Hall of India, 2001
2. Raj Kamal, Digital Systems Principles and Design, Pearson Education, First Edition, 2007

3. Charles H. Roth, Jr. and Larry L. Kinney, Fundamentals of Logic Design, CL Engineering, Seventh Edition, 2013.

Reference books

1. W. H. Gothmann, Digital Electronics -An Introduction to Theory and Practice, Prentice Hall of India, 2000
2. Donald D. Givone, Digital Principles and Design, Tata McGraw –Hill, Thirteenth Impression, 2003.

BCS-12 PRINCIPLES OF DATA STRUCTURES THROUGH C/C++

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Describe how arrays, records, linked lists, stacks, queues, trees, and graphs are represented in memory, used by the algorithms and their common applications.
2. Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs.
3. Compare and contrast the benefits of dynamic and static data structures implementations.
4. Identity the alternative implementations of data structures with respect to its performance to solve a real world problem.
5. Demonstrate organization of information using Trees and Graphs and also to perform different operations on these data structures.
6. Design and implement an appropriate organization of data on primary and secondary memories for efficient its efficient retrieval. .
7. Discuss the computational efficiency of the principal algorithms for sorting, searching and hashing.
8. Describe the concept of recursion, its application, its implementation and removal of recursion.

Topics Covered

UNIT-I

Introduction: Basic Terminology, Elementary Data Organization, Structure Operations, Complexity and Time-Space Tradeoff 9

Arrays: Definition, Representation and Analysis, Single and Multi Dimension Array, Address Calculation, Application of Arrays, Character, String in C, Character String Operation, Arrays Parameters, Ordered List, Sparse Matrices and Vectors

Stacks: Array Representation and Implementation of Stack, Operations on Stacks: Push &Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, Application of Stack, Conversion of Infix to Prefix and Postfix Expressions, Evaluation of Postfix Expressions using Stack, Application of Recursion in Problem like Tower of Hanoi

UNIT-II

Queues: Array and Linked Representation and Implementation of Queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular Queues, D-Queues and Priority Queues. 9

Linked List: Representation and Implementation of Singly Linked Lists, Two-Way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and Deletion to / from Linked Lists, Insertion and Deletion Algorithms, Doubly Linked List, Linked List in Array, Polynomial Representation and Addition, Generalized Linked List, Garbage Collection and Compaction.

UNIT-III

Trees: Basic Terminology, Binary Trees, Binary Tree Representation, Algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary Trees, Traversing Binary Trees, Threaded Binary Trees, Traversing Threaded Binary Trees, Huffman Algorithm. 9

Binary Search Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees, B-Trees.

UNIT-IV

Searching and Hashing: Sequential Search, Binary Search, Comparison and Analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation. 9

Sorting: Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort, Sorting on Different Keys, Practical Consideration for Internal Sorting.

Graphs: Terminology &Representations, Graphs &Multi-Graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

EXPERIMENTS

Write C/C++ Programs to illustrate the concept of the following:

1. Sorting Algorithms-Non-Recursive
2. Sorting Algorithms-Recursive
3. Searching Algorithm
4. Stack
5. Queue
6. Linked List
7. Graph

Textbooks

1. Horowitz and Sahani, Fundamentals of Data Structures, Galgotia Publication, New Delhi.
2. R. Kruseetal, Data Structure and Pragram Design in C, Pearson Education Asia Delhi
3. A. M.Tenenbaum, Data Structures using C & C++, PHI, India
4. K Loudon, Mastering Algorithms with C, Shroff Publication and Distributor Pvt. Ltd.
5. Bruno R Preiss, Data Structure and Algorithms with Object Oriented Design Pattern in C++, John Wiley & Sons
6. Adam Drozdek, "Data Structures and Algorithms in C++", Thomson Asia Pvt. Ltd. Singapore

Reference books

1. Lewis, H.R., Denenberg, L., Data Structures and their Algorithms. Published by Addison-Wesley, UK, 1991
2. Oluwadare, S.A., Agbonifo, O.C., Fundamentals of Data structures and Algorithms. Lecture Notes, 2013

BCS-13 INTERNET & JAVA PROGRAMMING

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. To identify different components of client server architecture on Internet computing.
2. Knowledge of how to develop and deploy applications and applets in JAVA.
3. Knowledge of how to develop and deploy GUI using JAVA Swing and AWT.
4. Design, develop and implement interactive web applications.
5. Be able to implement, compile, test and run JAVA programs comprising more than one class and to address a particular software problem.
6. To understand the basic concepts of Internet services and related technologies.
7. Develop programs using the JAVA Collection API as well as the JAVA standard class library.

Topics Covered

UNIT-I

Internet: Internet, Connecting to Internet: Telephone, Cable, Satellite Connection, Choosing an ISP, Introduction to Internet Services, E-Mail Concepts, Sending and Receiving Secure E-Mail, Voice and Video Conferencing. 9

UNIT-II

Core JAVA: Introduction, Operator, Data type, Variable, Arrays, Control Statements, Methods & Classes, Inheritance, Package and Interface, Exception Handling, Multithread Programming, I/O, JAVA Applet, String Handling, Networking, Event Handling, Introduction to AWT, AWT Controls, Layout Managers. 9

UNIT-III

JAVA Swing: Creating a Swing Applet and Application, Programming using Panes, Pluggable Look and feel, Labels, Text Fields, Buttons, Tabbed Panes. 9

JDBC: Connectivity Model, JDBC/ODBC Bridge, JAVA. SQL Package, Connectivity to Remote Database.

UNIT-IV

JAVA Beans: Application Builder Tools, The Bean Developer Kit(BDK), JAR files, Introspection, Developing a Simple Bean, using Bound Properties, The JAVA Beans API, Session Beans, Entity Beans, Introduction to JAVA Servlet: Servlet Basics, Servlet API Basic, Life Cycle of a Servlet, Running Servlet. 9

EXPERIMENTS

1. Basic programs of simple statements, conditional statements, iterative statements and arrays
2. Programs having object oriented concepts like Inheritance and Interface
3. Programs for Exception Handling and Event Handling
4. Programs of Threads and Multithreading
5. Programs related to Applets and Swings
6. Programs including JAVA Beans and Servlets

Textbooks

1. Naughton, Schildt, The Complete Reference JAVA2, TMH.
2. Balaguruswamy E, Programming in JAVA, TMH

Reference books

1. Margaret Levine Young, The Complete Reference Internet, TMH.
2. Dustin R. Callway, Inside Servlets, Addison Wesley.
3. Mark Wutica, JAVA Enterprise Edition, QUE.
4. Steven Holzner, JAVA2 Black book, Dreamtech.

BCS-14 SOFTWARE LAB-III

Course Category : Engineering Fundamental (EF)
Pre-requisite Subject : NIL
Contact Hours/Week : Lecture : 0, Tutorial : 0 , Practical: 4
Number of Credits : 2
Course Assessment : Continuous assessment through three Viva voce, Practical

Methods work/record, attendance and Major Practical Examination

Course Outcomes : The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Analyze and represent problems in the object-oriented programming paradigm.
2. Design and implement object-oriented software systems.
3. Demonstrate the efficient implementation of various Data Structures in memory and their operation.
4. Build programs on fundamental algorithmic problems including Searching, Sorting, Tree Traversals, Graph traversals, and shortest paths.
5. Explain the main principles for client-server programming
6. Design and implement Client-side systems, server-side system and event-driven graphical user interface.
7. Integrate their knowledge and skills to produce a real life application.

EXPERIMENTS

C++:

1. Program using functions with default arguments implementation of call by value, address, reference
2. Simple classes for understanding objects, member functions & constructors classes with primitive data members, classes with arrays as data members classes with pointers as data members classes with constant data members classes with static member functions
3. Compile time polymorphism- operator overloading, function overloading
4. Run time polymorphism -inheritance ,virtual functions, virtual base classes
5. File handling -sequential access, random access

JAVA:

1. Simple JAVA applications for understanding references to an instant of a class, handling strings in JAVA, simple package creation, developing user defined packages in JAVA
2. Interfaces
3. Threading- creation of threading in JAVA applications, multi-threading
4. Exception handling mechanism in JAVA- handling predefined exceptions, handling user defined exceptions

Internet Programming:

1. Web page creation using HTML
   - i) To embed an image map in a web page
   - ii) To fix the hot spots
   - iii) Show all the related information when the hot spots are clicked.
2. Web page creation with all types of Cascading style sheets
3. Client side scripts for validating web form controls using DHTML
4. JAVA programs to create applets

5. i) Create a color palette with matrix of buttons
    i) ii) Set background and foreground of the control text area by selecting a color from color palette.
    ii) In order to select foreground or background use check box control as radio buttons.
    iii) To set background images.

6. Programs in JAVA using servlets
7. Programs in JAVA to create three-tier applications using JSP and Databases
    i. for conducting online examination
    ii. for displaying students mark list.
8. Programs using XML-schema-XSLT/XSL
9. Programs using AJAX
10. Implementation of web services and databases.

BCS-15 DATABASE MANAGEMENT SYSTEMS

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. List and define the fundamental concepts of database management system.
2. Manually execute a given (simple) database design a transaction over it.
3. Manually infer the type of a given (simple) database transaction.
4. Implement (simple) algorithms and data structures as database transaction.
5. Design (large) databases that are modular and have reusable components.
6. Explain on a simple problem how to apply concurrency control over concurrent database transactions.

Topics Covered

UNIT-I

Introduction: An Overview of Database Management System, Database System vs File System, 9

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 Modeling using 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.

UNIT-II

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. 9

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

UNIT-III

Database Design &Normalization: Functional Dependencies, Normal Forms, First, Second, Third Normal Forms, BCNF, Inclusion Dependence, Loss Less Join Decompositions, Normalization using FD, MVD, and JDS, Alternative Approaches to Database Design. 9

UNIT-IV

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. 9

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.

EXPERIMENTS

1. Exercises to be based on Sybase / Oracle / Postgres / VB / Power Builder / DB2 / MS-Access.
2. Applications involving vendor development systems, stores management system, finance management etc.
3. Creation and querying of database tables for following cases. .
   i. Write SQL queries using logical operations (=,<,>,etc)
   ii. Write SQL queries using SQL operators
   iii. Write SQL query using character, number, date and group functions
   iv. Write SQL queries for relational algebra
   v. Write SQL queries for extracting data from more than one table
   vi. Write SQL queries for sub queries, nested queries
   vii. Write program by the use of PL/SQL
   viii. Concepts for ROLL BACK, COMMIT & CHECK POINTS
   ix. Create VIEWS, CURSORS and TRGGERS & write ASSERTIONS.
   x. Create FORMS and REPORTS
4. Design of tables by normalization and dependency analysis.

5. Writing application software with host language interface

Textbooks

1. Date C J, An Introduction to Database Systems, Addison Wesley
2. Korth, Silbertz, Sudarshan, Database Concepts, McGraw Hill
3. Elmasri, Navathe, Fundamentals of Database Systems, Addison Wesley
4. O'Neil, Databases, Elsevier Pub.
5. Leon& Leon, Database Management Systems, Vikas Publishing House
6. Bipin C. Desai, An Introduction to Database Systems, Galgotia Publications
7. Majumdar & Bhattacharya, Database Management System, TMH
8. Ramkrishnan, Gehrke, Database Management System, McGraw Hill
9. Kroenke, Database Processing Fundamentals, Design and Implementation, Pearson Education.
10. J. D. Ulman, Principles of Database and Knowledge base System, Computer Science Press.
11. Maheshwari Jain. DBMS: Complete Practical Approach, Firewall Media, New Delhi

Reference books

1. Ramon a. Mato-Toledo, Pauline K. Cushman, Database Management Systems, Schaums' Outline series, TMH, New Delhi Special Indian Edition 2007
2. Ivan Bayross, Mastering Database Technologies, BPB Publications, New Delhi - First Indian Edition 2006, Reprinted 2011

BCS-16 THEORY OF COMPUTATION

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous Assessment through Tutorials, Assignments, Quizzes and Three Minor Tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Analyse and design finite automata, pushdown automata, Turing machines, formal languages, and grammars.
2. Demonstrate the understanding of key notions, such as algorithm, computability, decidability, and complexity through problem solving.
3. Prove the basic results of the Theory of Computation.
4. State and explain the relevance of the Church-Turing thesis.

Topics Covered

UNIT-I

Alphabets, Strings and Languages, Automata and Grammars, Deterministic Finite Automata (DFA)-Formal Definition, Simplified Notation: State Transition Graph, Transition Table, Language of DFA, Nondeterministic Finite Automata (NFA), NFA with Epsilon Transition, Equivalence of NFA and DFA, Minimization of Finite Automata, Myhill-Nerode Theorem 9

UNIT-II

Regular Expression (RE), Definition, Operators of Regular Expression and their Precedence, Algebraic Laws for Regular Expressions, Kleen's Theorem, Regular Expression to FA, DFA to Regular Expression, Arden Theorem, Non Regular Languages, Pumping Lemma for Regular Languages. Application of Pumping Lemma, Closure Properties of Regular Languages, Decision Properties of Regular Languages, FA with Output: Moore and Mealy Machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA. 9

UNIT-III

Context Free Grammar (CFG) and Context Free Languages (CFL): Definition, Examples, Derivation, Derivation Trees, Ambiguity in Grammer, Inherent Ambiguity, Ambiguous to Unambiguous CFG, Useless Symbols, Simplification of CFGs, Normal Forms for CFGs: CNF and GNF, Closure Proper ties of CFLs, Decision Properties of CFLs: Emptiness, Finiteness and Membership, Pumping Lemma for CFLs. 9

Push Down Automata (PDA): Description and Definition, Instantaneous Description, Language of PDA, Acceptance by Final State, Acceptance by Empty Stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two Stack PDA

UNIT-IV

Turing Machines (TM): Basic Model, Definition and Representation, Instantaneous Description, Language Acceptance by TM, Variants of Turing Machine, TM as Computer of Integer Functions, Universal TM, Church's Thesis, Recursive and Recursively Enumerable Languages, Halting Problem, Introduction to Undecidability, Undecidable Problems about TMs. Post Correspondence Problem (PCP), Modified PCP, Introduction to Recursive Function Theory. 9

Textbooks

1. Micheal Sipser, "Introduction to the Theory of Computation", Thomson Learning

Reference books

1. Hopcroft, Ullman, "Introduction to Automata Theory, Languages and Computation", Pearson Education
2. Peter Linz, "An Introduction to Formal Language and Automata", Narosa Publishing house
3. H R. Lewis and Christos H. Papadimitriou, "Elements of the theory of Computation", PHI Ltd

BCS-17 COMPUTER ORGANIZATION & DESIGN

Course Category : Department Core (DC)
Pre-requisite Subject : NIL
Contact Hours/Week : Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits : 5
Course Assessment : Continuous assessment through tutorials, attendance, home

Methods		assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. To understand the basic structure and operation of digital computer.
2. To study the design of arithmetic and logic unit and implementation of fixed point and floating-point arithmetic operations
3. To study the two types of control unit techniques and the concept of Pipelining
4. To study the hierarchical memory system including cache memories and virtual memory
5. To study the different ways of communicating with I/O devices and standard I/O interfaces

Topics Covered

UNIT-I

Register Transfer Language, Bus and Memory Transfers, Bus Architecture, Bus Arbitration, Arithmetic Logic, Shift Micro-Operation, Arithmetic Logic Shift Unit, Design of Fast Address, IEEE Standard for Floating Point Numbers. 9

UNIT-II

Control Design: Hardwired &Micro Programmed Control Unit, .Processor Design: Processor Organization: General Register Organization, Stack Organization, Addressing Mode, Instruction Format, Data Transfer & Manipulations, Program Control, Reduced Instruction Set Computer, Pipelining 9

UNIT-III

Arithmetic - Addition & Subtraction of Signed Numbers - Multiplication - Integer Division - Floating Point Operations, Decimal Arithmetic Unit, Decimal Arithmetic Operations. 9

UNIT-IV

Input-Output Organization: I/O Interface, Modes of Transfer, Interrupts &Interrupt Handling, Direct Memory Access, Input-Output Processor, Serial Communication. 9

Memory Organization: Memory Hierarchy, Main Memory (RAM and ROM Chips), Auxiliary Memory, Cache Memory, Virtual Memory

EXPERIMENTS

1. Implementing HALF ADDER, FULL ADDER using basic logic gates
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8 line DECODER.
4. Implementing 4x1 and 8x1 MULTIPLEXERS.
5. Verify the excitation tables of various FLIP-FLOPS.
6. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
7. Design of an 8-bit ARITHMETIC LOGIC UNIT.
8. Design the data path of a computer from its register transfer language description.
9. Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
10. Implement a simple instruction set computer with a control unit and a data path.

Textbooks

1. Computer System Architecture - M. Mano
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.

Reference books

1. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006.
2. Behrooz Parahami, "Computer Architecture", Oxford University Press, Eighth Impression, 2011.
3. David A. Patterson and John L. Hennessy, "Computer Architecture-A Quantitative Approach", Elsevier, a division of reed India Private Limited, Fifth edition, 2012.

BCS-18 SOFTWARE LAB-IV

Course category	:	Engineering Fundamental (EF)
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture : 0, Tutorial : 0 , Practical: 4
Number of Credits	:	2
Course Assessment methods	:	Continuous assessment through three Viva voce, Practical work/record, attendance and Major Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Tools and techniques for optimizations in design processes.
2. Design and develop the software packages/ systems to support the management of an organization in question.

3. Design and develop a DBMS.

EXPERIMENTS

1. Write user-friendly computer programs to implement algorithms in your course of Optimization Techniques.
2. Design and develop a software packages/ systems for your University Management System.
3. Design and develop your own DBMS.
4. Design and develop a simulator for (i) Logic Circuit Design, (ii) Electronic Circuit Design.

BCS-19 WEB DESIGNING

Course Category	:	For other Department
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 0, Tutorial : 0 , Practical: 4
Number of Credits	:	2
Course Assessment Methods	:	Continuous Assessment through Practical Work/ Attendance/ Record/ Viva Voce, Three Viva Voce Examinations and One Major Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Identify common design mistakes when creating a web based application.
2. Discuss the process of editing a web page using text editors and web page editors.
3. Cover commonly used HTML tags and discuss how this knowledge is important to a web designer
4. Demonstrate an understanding of basic CSS, XML

EXPERIMENTS

1. Create a HTML static web page which shows the use of different tags in that.
2. Insert an image and create a link such that clicking on image takes user to other page.
3. Prepare a sample code to illustrate three types of lists in HTML.
4. Use tables to provide layout to your HTML page describing your university infrastructure.
5. Use frames such that page is divided into 3 frames 20% on left to show contents of pages, 60% in center to show body of page, remaining on right to show remarks.
6. Create a simple form that will show all the INPUT METHODS available in HTML.
7. Create a sample code to illustrate the Embedded, External and Inline style sheets for your web page.
8. Write an XML example of given tree that demonstrates the creation of user-designed tags and display it in a browser. fname, lname, joindate, bdate, college, employee, age, salary (with at least 3 elements) .
9. Write a program in XML for creation of DTD which specifies a particular set of rules.
10. Create an e-book having left side of the page name of the chapters and right side of the page the contents of the chapters clicked on left side.

Textbooks

1. Uttam K. Roy,**Web Technologies, 1/e ,** Oxford University Press, USA
2. Murray, Tom/Lynchburg, Creating a Web Page and Web Site ,College,2002
3. A beginner's guide to HTML NCSA,14th May,2003
4. Kogent Learning Solutions Inc. HTML 5 in simple steps Dreamtech Press

Reference books

1. Steven M. Schafer HTML, XHTML, and CSS Bible, 5ed ,Wiley India
2. Kogent Learning Web Technologies: HTML, JAVA script , Wiley

BCS-26 PRINCIPLES OF OPERATING SYSTEMS

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand the structure and functions of OS.
2. Learn about Processes, Threads and Scheduling algorithms.

3. Understand the principles of concurrency and Deadlocks.
4. Learn various memory management scheme.
5. Study I/O management and File systems.

Topics Covered

UNIT-I

Operating Systems Overview-Components, Goals of Designer, System Structures, User Services, Interrupt Systems and Device Programming-Interrupt Sources and Priorities, Interrupt Service Routines, Hardware Support - Machine States, Context Switching, Privileged Instructions and Registers 9

UNIT-II

Memory Management-Major Issues: Fetch, Placement, Contiguity, Relocation Adjustment, Paging and Virtual Memory, Translate-Look-Aside Buffer (Associative Memory), Single and Multi-Level Page Tables, Paging with Segmentation, Problems of Large Address Spaces and How They Are Addressed 9

Virtual Storage Management- Storage Hierarchy, Cache Usage, Partial Residency, Page Replacement Strategies, Working Sets

UNIT-III

Concurrency Problems and Solutions- Critical Section Problem, Process Synchronization and Coordination, Semaphores, Special Instructions, Monitors, Inter-process Communication, Remote Procedure Calls, Special Problems of Transaction-Based Systems 9

Deadlock and Resource Conflict- Prevention, Avoidance, Detection, Recovery,

Process and Thread Management-Process/Thread Creation and Termination, Process/Thread States and Their Transitions

CPU Scheduling Algorithms, Non-Preemptive Approaches, Preemptive Approach, Multi-Processor Considerations

UNIT-IV

Physical Storage Management- Disk Scheduling Algorithms, Disk Performance Features, Disk Reliability Concerns 9

File System Organization - The Boot Record - Where Things Start, Directory Organization, File Descriptors, Access Control Backup

System Security-Principle of Least Privilege, Threats and Vulnerabilities, Protection Mechanisms - Access and Capability Control, User (Subject) Authentication, Levels of Security in "Trusted" Systems, Confinement Problem

EXPERIMENTS

1. Study of hardware and software requirements of different operating systems (UNIX,LINUX,WINDOWS XP, WINDOWS7/8
2. Execute various UNIX system calls for
   a. Process management
   b. File management
   c. Input/output Systems calls
3. Implement CPU Scheduling Policies:
   a. SJF

b. Priority
c. FCFS
d. Multi-level Queue

4. Implement file storage allocation technique:
   a. Contiguous(using array)
   b. Linked –list(using linked-list)
   c. Indirect allocation (indexing)
5. Implementation of contiguous allocation techniques:
   a. Worst-Fit
   b. Best- Fit
   c. First- Fit
6. Calculation of external and internal fragmentation
   a. Free space list of blocks from system
   b. List process file from the system
7. Implementation of compaction for the continually changing memory layout and calculate total movement of data
8. Implementation of resource allocation graph RAG)
9. Implementation of Banker's algorithm
10. Conversion of resource allocation graph (RAG) to wait for graph (WFG) for each type of method used for storing graph.
11. Implement the solution for Bounded Buffer (producer-consumer)problem using inter process communication techniques-Semaphores
12. Implement the solutions for Readers-Writers problem using inter process communication technique -Semaphore

Textbooks

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, John Wiley & Sons (ASIA) Pvt. Ltd, Seventh Edition, 2005
2. Pramod Chandra and P. Bhatt, "An Introduction to Operating Systems Concepts and Practice", Prentice Hall India,3rd Edition,2010

Reference books

1. Milenekovie, Operating System Concept, McGraw Hill.
2. Harvey M. Deitel, Paul J. Deitel, and David R. Choffnes, Operating Systems, Prentice Hall, Third edition, 2003
3. Petersons, "Operating Systems", Addison Wesley
4. Andrew S. Tannenbaum & Albert S. Woodhull, "Operating System Design and Implementation", Prentice Hall , 3rd Edition, 2006
5. William Stallings, Operating Systems – internals and design principles, Prentice Hall, 7thEdition, 2011
6. Gary J. Nutt, "Operating Systems", Pearson/Addison Wesley, 3rd Edition 2004.
7. Andrew S. Tannenbaum, "Modern Operating Systems", Prentice Hall,3rd Edition,2007.

BCS-27 COMPUTER GRAPHICS

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Have a basic understanding of the core concepts of computer graphics.
2. Be capable of using OpenGL to create interactive computer graphics.
3. Understand a typical graphics pipeline.
4. Have made pictures with their computer.

Topics Covered

UNIT-I

BASICS OF COMPUTER GRAPHICS- Introduction, Area of Computer Graphics, Design and Drawing, Animation Multimedia Applications, Simulation, How are Pictures Actually Stored and Displayed, Difficulties for Displaying Pictures. 9

GRAPHIC DEVICES- Cathode Ray Tube, Quality of Phosphors, CRTs for Color Display, Beam Penetration CRT, Shadow - Mask CRT, Direct View Storage Tube, Tablets, Light Pen, Three Dimensional Devices. C Graphics Basics Graphics Programming, Initializing Graphics, C Graphical Functions, Simple Programs.

SIMPLE LINE DRAWING METHODS- Point Plotting Techniques, Qualities of Good Line Drawing Algorithms, Digital Differential Analyzer(DDA), Bresenham's Algorithm, Generation of Circles

UNIT-II

TWO DIMENSIONAL TRANSFORMATIONS and CLIPPING AND WINDOWING- What is Transformation?, Matrix Representation of Points, Basic Transformation, Need for Clipping and Windowing, Line Clipping Algorithms, Midpoint Subdivision Method, Other Clipping Methods, Sutherland - Hodgeman Algorithm, Viewing Transformations. 9

GRAPHICAL INPUT TECHNIQUES- Graphical Input Techniques, Positioning Techniques, Positional Constraints, Rubber Band Techniques.

EVENT HANDLING AND INPUT FUNCTIONS- Introduction, Polling, Event Queue, Functions for Handling Events, Polling Task Design, Input Functions, Dragging and Fixing, Hit Detection, OCR.

UNIT-III

THREE DIMENSIONAL GRAPHICS- Need for 3-Dimensional Imaging, Techniques for 3-Dimesional Displaying, Parallel Projections, Perspective Projection, Intensity Cues, Stereoscope 9

Effect, Kinetic Depth Effect, Shading.

CURVES AND SURFACES- Shape Description Requirements, Parametric Functions, Bezier Methods, Bezier Curves, Bezier Surfaces, B-Spline Methods

UNIT-IV

SOLID AREA SCAN CONVERSION-Three Dimensional Transformations Solid Area Scan Conversion, Scan Conversion of Polygons, Algorithm Singularity, Three Dimensional Transformation, Translations, Scaling, Rotation, Viewing Transformation, Perspective, Algorithms, Three Dimensional Clipping, Perspective View of Cube. 9

HIDDEN SURFACE REMOVAL-Need For Hidden Surface Removal, Depth - Buffer Algorithm, Properties that Help in Reducing Efforts, Scan Line Coherence Algorithm, Span - Coherence Algorithm, Area-Coherence Algorithms, Warnock's Algorithm, Priority Algorithms

EXPERIMENTS

Develop program to

1. Understand the basic concepts of computer graphics.
2. Design scan conversion problems using C/C++ programming.
3. Apply clipping and filling techniques for modifying an object.
4. Understand the concepts of different type of geometric transformation of objects in 2D and 3D.
5. Understand the practical implementation of modeling, rendering, viewing of objects.

Textbooks

1. Z. Xiang, R. Plastock, Schaum's outlines Computer Graphics, 2nd Ed., TMH
2. B M Havaldar, C Graphics & Projects, Anmol Publications Pvt. Limited, 01-Jan-2005
3. Hearn and Baker Computer Graphics with OpenGL, 3e, Prentice Hall, 2004.
4. Asthana and Sinha, Computer Graphics for Scientists and Engineers, New Age International, 01-Jan-2007

Reference books

1. Foley, Vandam, Feiner, Hughes, Computer Graphics principles, 2ndEd.,Pearson Education
2. W. M. Newman, R. F. Sproull, Principles of Interactive computer Graphics, TMH.

BCS-28 DESIGN & ANALYSIS OF ALGORITHMS

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Define the basic concepts of algorithms and analyze the performance of algorithms.
2. Discuss various algorithm design techniques for developing algorithms.
3. Discuss various searching, sorting and graph traversal algorithms.
4. Understand NP completeness and identify different NP complete problems.
5. Discuss various advanced topics on algorithm

Topics Covered

UNIT-I

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. Divide And Conquer with Examples such as Sorting, Matrix Multiplication, Convex Hull and Searching. 9

UNIT-II

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. 9

Dynamic Programming with Examples such as Multistage Graphs, Knapsack, All Pair Shortest Paths -Warshal's and Floyd's Algorithms, Resource Allocation Problem.

UNIT-III

Backtracking, Branch and Bound with Examples such as Travelling Salesman Problem, Graph Coloring, N-Queen Problem, Hamiltonian Cycles and Sum Of Subsets 9

Advanced Data Structures***:*** Red-Black Trees, B – Trees, Binomial Heaps, Fibonacci Heaps.

UNIT-IV

Selected Topics: String Matching, Text Processing- Justification of Text, Theory of NP-Completeness, Approximation Algorithms And Randomized Algorithms, Algebraic Computation, Fast Fourier Transform. 9

EXPERIMENTS

1. To analyze time complexity of Insertion sort.
2. To analyze time complexity of Quick sort.
3. To analyze time complexity of Merge sort.
4. To Implement Largest Common Subsequence.
5. To Implement Matrix Chain Multiplication.
6. To Implement Strassen's matrix multiplication Algorithm, Merge sort and Quick sort.
7. To implement Knapsack Problem.
8. To implement Activity Selection Problem.
9. To implement Dijkstra's Algorithm.
10. To implement Warshall's Algorithm.
11. To implement Bellman Ford's Algorithm.
12. To implement Naïve String Matching Algorithm.
13. To implement Rabin Karp String Matching Algorithm
14. To implement Prim's Algorithm.
15. To implement Kruskal's Algorithm.

Textbooks

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, Introduction to Algorithms, PHI.
2. RCT Lee, SS Tseng, RC Chang and YT Tsai, "Introduction to the Design and Analysis of Algorithms", McGraw Hill, 2005.
3. Ellis Horowitz and Sartaj Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, Maryland, 1978
4. Berman, Paul," Algorithms", Cengage Learning.
5. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.

Reference books

1. Berlion, P. Izard, P., Algorithms-The Construction, Proof and Analysis of Programs, 1986. Johan Wiley & Sons.
2. Bentley, J.L., Writing Efficient Programs, PHI
3. Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasekaran, *Computer Algorithms*, W. H. Freeman, NY, 1998
4. Goodman, S.E. & Hedetnien, introduction to Design and Analysis of Algorithm1997, MGH.
5. Knuth, D.E , Fundamentals of Algorithms: The Art of Computer Programming Vol,1985

BCS-29 ADVANCED COMPUTER ARCHITECTURE

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and

Three Minor tests and One Major Theory & Practical Examination

Course Outcomes : The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand the advanced concepts of computer architecture.
2. Exposing the major differentials of RISC and CISC architectural characteristics.
3. Investigating modern design structures of Pipelined and Multiprocessors systems.
4. Become acquainted with recent computer architectures and I/O devices, as well as the low-level language required to drive/manage these types of advanced hardware.
5. Preparing selected reports that imply some emergent topics supporting material essence.

Topics Covered

UNIT-I

RISC Processors, Characteristics of RISC Processors, RISC vs CISC, Classification of Instruction Set Architectures, Review of Performance Measurements, Basic Parallel Processing Techniques: Instruction Level, Thread Level and Process Level, Classification of Parallel Architectures. 9

UNIT-II

Basic Concepts of Pipelining, Arithmetic Pipelines, Instruction Pipelines, Hazards in A Pipeline: Structural, Data, and Control Hazards, Overview of Hazard Resolution Techniques, Dynamic Instruction Scheduling, Branch Prediction Techniques, Instruction-Level Parallelism using Software Approaches, Superscalar Techniques, Speculative Execution. 9

UNIT-III

Basic Concept of Hierarchical Memory Organization, Main Memories, Cache Design and Optimization, Virtual Memory Design and Implementation, Memory Protection, Evaluating Memory Hierarchy Performance, RAID, Centralized vs. Distributed Shared Memory. 9

UNIT-IV

Interconnection Topologies, Synchronization, Memory Consistency, Review of Modern Multiprocessors, Distributed Computers, Clusters, Grid, Mainframe Computers, Bus Structures and Standards, Types and Uses of Storage Devices, Interfacing I/O to The Rest of the System, Reliability and Availability, I/O System Design 9

EXPERIMENTS

1. Write an algorithm and program to perform matrix multiplication of two n * n matrices on the 2-D mesh SIMD model.
2. Write an algorithm and program to perform matrix multiplication of two n * n matrices on Hypercube SIMD Model
3. Write an algorithm and program for Block oriented Matrix Multiplication on multiprocessor system
4. Study of Scalability for Single board Multi-board, multi-core, multiprocessor using Simulator
5. Study of various computer Architecture (MIPS, Power etc.) using simulator.
6. Study of Memory and system controllers, Interrupt and DMA controllers using simulator.

Textbooks

1. Hennessey and Patterson, Computer Architecture: A quantitative Approach, Morgan Kaufman.
2. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.
3. SIMA, Advanced Computer Architectures, Addison-Wesley.

Reference books

1. H.S. Stone, High-performance Computer Architecture, 3rd edition, Addison-Wesley, 1993.
2. Patterson, D. A. and Hennessy, J. L., Computer Organization and Design: The Hardware/ Software Interface, Morgan Kaufmann, 1998.

BCS-30 SEMINAR

Course category	:	Audit Course (AC)
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture : 0, Tutorial : 0 , Practical: 6
Number of Credits	:	3
Course Assessment methods	:	Continuous assessment through quality of material, presentation, quality & extent of external response of question asked and participation in other seminars (attendance)
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. To expose students to the 'real' working environment and get acquainted with the organization structure, business operations and administrative functions.
2. Students will demonstrate the ability to distinguish opinions and beliefs from researched claims and evidence and recognize that kinds of evidence will vary from subject to subject.
3. Students will demonstrate the ability to evaluate, credit, and synthesize sources.

BCS-31 PRINCIPLE OF COMPILER DESIGN

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Define the phases of a typical compiler, including the front--- and back--- end.
2. Identify tokens of a typical high---level programming language; define regular expressions for tokens and design; implement a lexical analyzer using a typical scanner generator.
3. Explain the role of a parser in a compiler and relate the yield of a parse tree to a grammar derivation; design and implement a parser using a typical parser generator.
4. Apply an algorithm for a top---down or a bottom---up parser construction; construct a parser for a small context---free grammar.
5. Explain the role of a semantic analyzer and type checking; create a syntax---directed definition and an annotated parse tree; describe the purpose of a syntax tree.
6. Explain the role of different types of runtime environments and memory organization for implementation of typical programming languages.
7. Describe the purpose of translating to intermediate code in the compilation process.
8. Design and implement an intermediate code generator based on given code patterns

Topics Covered

UNIT-I

CompilerStructure:Analysis-SynthesisModelofCompilation,Various Phases of A Compiler, Tool Based Approach to Compiler Construction Lexical Analysis: Interface with Input, Parser and Symbol Table, Token, Lexeme and Patterns, Difficulties in Lexical Analysis, Error Reporting, and Implementation. Regular Definition, Transition Diagrams, LEX. 9

UNIT-II

Syntax Analysis: Context Free Grammars, Ambiguity, Associativity, Precedence, Top Down Parsing, Recursive Descent Parsing, Transformation on the Grammars, Predictive Parsing, Bottom Up Parsing, Operator Precedence Grammars, LR Parsers (SLR, LALR, LR), YACC. 9

UNIT-III

Syntax Directed Definitions: Inherited and Synthesized Attributes, Dependency Graph, Evaluation Order, Bottom Up and Top Down Evaluation Of Attributes, L- and S-Attributed Definitions. 9

Type Checking: Type System, Type Expressions, Structural and Name Equivalence of Types, Type Conversion, Overloaded Functions and Operators, Polymorphic Functions.

Intermediate Code Generation: Intermediate Representations, Translation of Declarations, Assignments Intermediate Code Generation For Control Flow, Boolean Expressions and Procedure Calls, Implementation Issues.

UNIT-IV

Symbol Table Management, Runtime Environments, Source Language Issues, Storage Organization, Storage Allocation Strategies, Access to Non-Local Names, Parameter Passing. 9

Code Optimization, Peephole Optimization, Source of Optimizations, Optimization of Basic Blocks, Loops, Global Dataflow Analysis, Introduction to Code Generation.

EXPERIMENTS

1. Write a program using Lex to calculate the number of characters, number of words and the number of lines present in the given text file as input.
2. Write a program using Lex to implement the set of regular expression and indicates the acceptance of a given string for a particular regular expression.
3. Write a C program to implement the conversion of regular expression to non-deterministic finite automation
4. Write a program using Yacc to check whether a string belong to the given grammar or not.
5. Write a C program to compute FIRST and FOLLOW of the non-terminals of given grammar.
6. Write a C program to check the given grammar is Left recursive and remove Left recursion.
7. Write Syntax Directed Translation actions using Yacc to generate Parse Tree for the grammar for arithmetic expressions.
8. Write Syntax Directed Translation actions using Yacc to translate arithmetic expressions into Post-fix form.
9. Write Syntax Directed Translation actions using Yacc to translate arithmetic expressions into three address code.

Textbooks

1. A.V. Aho, M.S. Lam, R. Sethi, and J.D. Ullman, Compilers: Principles, Techniques, and Tools, Pearson Education, 2007 (second ed.).
2. K.D. Cooper, and L. Torczon, Engineering a Compiler, Elsevier, 2004.

Reference books

1. AW Appel, J Palsberg, Modern Compiler Implementation in JAVA, Cambridge University Press, 2002
2. AW Appel, M Ginsburg, Modern Compiler Implementation in C, Cambridge University Press.

BCS-32 ARTIFICIAL INTELLIGENCE

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. The intelligent agents--software or hardware entities that perform useful tasks with some degree of autonomy.
2. An understanding of the basic areas of artificial intelligence including problem solving, knowledge representation, reasoning, decision making, planning, perception and action, and learning -- and their applications (e.g., data mining, information retrieval)
3. Design and implement key components of intelligent agents of moderate complexity in JAVA and /or Lisp or Prolog and evaluate their performance.
4. Develop familiarity with current research problems, research methods, and the research literature in AI

Topics Covered

UNIT-I

Artificial Intelligence Introduction, Intelligent Agents, Solving Problems by Searching Beyond Classical Search Adversarial Search Constraint Satisfaction Problems 9

UNIT-II

Knowledge and Reasoning Logical Agents First-Order Logic Inference in First-Order Logic Classical Planning and Acting in the Real World Knowledge Representation Uncertain Knowledge and Reasoning Quantifying Uncertainty Probabilistic Reasoning Probabilistic Reasoning over Time 16 Making Simple Decisions Making Complex Decisions 9

UNIT-III

Planning and Acting in the Real World Definition of Classical Planning Algorithms for Planning as State-Space Search Planning Graphs Classical planning as Boolean Satisfiability Representing temporal and resource constraints Planning and Acting in Nondeterministic Domains. Knowledge Representation Acting under Uncertainty Probabilistic Reasoning Time and Uncertainty Learning from Examples Knowledge in Learning Probabilistic Models Reinforcement Learning 9

UNIT-IV

Forms of Learning Supervised Learning, Decision Trees Evaluating and Choosing the Best Hypothesis A Logical Formulation of Learning Statistical Learning with Complete Data Natural Language Processing Communicating, Perceiving, and Acting Natural Language Processing Natural Language for Communication Perception Robotics 9

EXPERIMENTS

1. Write the program to solve the water jug problem using production rule set.
2. Write the program to solve the water jug problem using A* ALGORITHM.
3. Write the program to solve the 8 puzzle problem using A* ALGORITHM.
4. Write the program to solve the salesman problem using A* ALGORITHM.
5. Write the program to solve the farmer transfer three belonging form one side of the river to other side using AO* ALGORITHM.
6. Write the program to solve the DISEASE problem using Bayesian reasoning.
7. Write the program to solve the Object finding problem using Bayesian reasoning.
8. Write the program to solve the Object finding problem using D S theory
9. Write the program to solve the Decision Trees Evaluating.
10. Write the program for walk, drive, take the bus, take a cab, and fly problem using mean end analysis.

Textbooks

1. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach", Second Edition, Pearson Education, 2012

Reference books

2. David Poole, Alan Mackworth, Randy Goebel, Computational Intelligence: a logical approach, Oxford University Press, 2012.
3. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problem solving", Fourth Edition, Pearson Education, 2012
4. J. Nilsson, "Artificial Intelligence: A new Synthesis", Elsevier Publishers, 1998

BCS-33 WEB TECHNOLOGIES

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Identify common design mistakes when creating a web based application.
2. Discuss the process of editing a web page using text editors and web page editors
3. Cover commonly used HTML tags and discuss how this knowledge is important to a web designer
4. Demonstrate an understanding of basic CSS, XML, JAVA Script, JSP, ASP.NET and PHP

Topics Covered

UNIT-I

Introduction to WWW- World Wide Web, WWW Architecture, Web Search Engines, Web Crawling, Web Indexing, Web Searching, Search Engines Optimization and Limitations, Web Mining: Web Content Mining, Web Structure Mining, Web Usage Mining 9

UNIT-II

Markup Language Basics: SGML, HTML, CSS And XML 9

SGML: Standard Generalized Markup Language (SGML) -Structures, Elements, Content Models, DTD, Attributes Entities.

HTML: Designing Web Pages With HTML-Use Of Tags, Hyperlinks, URLs, Tables, Text Formatting, Graphics & Multimedia, Imagemap, Frames and Forms in Web Pages.

CSS: Use of Cascading Style Sheet in Web Pages.

XML: Extensible Markup Language (XML): Introduction using User-Defined Tags in Web Pages, Displaying XML Contents, XML Dtds, Use of XSL

UNIT-III

Client-Side Scripting using JAVA Script 9

JAVA script Overview; Constants, Variables, Operators, Expressions & Statements; User-Defined & Built-in Functions; Client-Side Form Validation; Using Properties and Methods of Built-in Objects

UNIT-IV

Server-Side Scripting Using JSP, **ASP.NET And PHP** 9

JSP : Introduction to JSP, JSP Architecture, JSP Directives, JSP Scripting Elements, Default Objects in JSP, JSP Actions, JSP with Beans and JSP with Database, Error Handling in JSP, Session Tracking Techniques in JSP, Introduction to Custom Tags.

ASP.NET : ASP.Net Coding Modules, ASP.NET Page Directives, Page Events and Page Life Cycle , Postback and Crosspage Posting ASP.NET Server Controls , HTML Controls, Validation Controls, Building Databases .

PHP(Hypertext Preprocessor)-Introduction, Syntax, Variables, Strings, Operators, If- Else, Loop, Switch, Array, Function, Form ,Mail, File Upload, Session, Error, Exception, Filter, PHP-ODBC

EXPERIMENTS

1. Create a HTML static web page which shows the use of different tags in that.
2. Insert an image and create a link such that clicking on image takes user to other page.
3. Prepare a sample code to illustrate three types of lists in HTML.
4. Use tables to provide layout to your HTML page describing your university infrastructure

5. Use frames such that page is divided into 3 frames 20% on left to show contents of pages, 60% in center to show body of page, remaining on right to show remarks.
6. Create a simple form that will show all the INPUT METHODS available in HTML.
7. Create a sample code to illustrate the Embedded, External and Inline style sheets for your web page.
8. Write down simple JAVA Script using timeout such that image will be changed after every 1 ms at a specified position.
9. Design a registration form and validate its field by using JAVA script.
10. Write an XML example of given tree that demonstrates the creation of user-designed tags and display it in a browser.
11. college, employee, fname, lname, joindate, bdate, age, salary (with atleast 3 elements)
12. Write a program in XML for creation of DTD which specifies a particular set of rules.
13. Create a bean student with attributes (first name, last name, age, class). In another JSP page display the bean values using <jsp:usebean>.
14. Write a program to use JDBC connectivity program for maintaining database by sending queries through JSP Page.
15. Use ad-rotator to change advertisements on client side request.(ASP.NET)
16. Implement Session tracking using user authentication in ASP.NET.
17. Write a PHP script to create a database Student DB.
18. Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.

Textbooks

1. Uttam K. Roy, **Web Technologies, 1/e,** Oxford University Press, USA
2. M. Srinivasan, Web Technology: Theory and Practice, Pearson Education India
3. Deitel, Deitel and Nieto, Internet and Worldwide Web - How to Program, 5th Edition, PHI, 2011.
4. Ralph Moseley & M. T. Savaliya , Developing Web Application- Second Edition, Wiley
5. Miller/Kirst, Web Programming Step by Step, Stepp, 2nd edition, 2009
6. Ullman , PHP for the Web: Visual Quick Start Guide, Pearson Education, 4th edition, 201
7. www.w3c.org
8. www.w3schools.com

Readings:

Various journal and conference articles, research reports, and book excerpts as appropriate

Reference books

1. Ivan Bayross , Web Enabled Commercial Application Development Using HTML, DHTML, JAVA Script, Perl & CGI, BPB Publication, 2005
2. Hans Bergsten, JAVA Server Pages, O'Reilly.

BCS-34 COMPUTER NETWORKS

Course Category : Department Core (DC)
Pre-requisite Subject : NIL
Contact Hours/Week : Lecture : 3, Tutorial : 1 , Practical: 2

Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Understand the concepts of communication architecture and protocols
2. Identify different types of communication mediums and techniques
3. Define and identify different types of multiplexing, data encoding, modulation, and switching techniques
4. Illustrate different standards of Local Area Network in terms of technologies and hardware used
5. Illustrate network addressing and analysis techniques
6. Understand the Wide Area Network technologies
7. Understand the network routing concepts
8. Understand the internetworking concepts and architectures
9. Understand the TCP/IP protocols and design architectures

Topics Covered

UNIT-I

Introductory Concepts: Goals and Applications of Networks, Network Structure and Architecture, OSI Reference Model, Services, Networks Topology, Physical Layer- Transmission, Switching Methods, LAN Inter Connection Devices, Integrated Services Digital Networks. 9

UNIT-II

Medium Access Sub Layer: Channel Allocations, LAN Protocols, ALOHA Protocols- Pure ALOHA, Slotted ALOHA, Carrier Sense Multiple Access Protocols, CSMA with Collision Detection, Collision Free Protocols, IEEE Standards, Ethernet, FDDI, Data Link Layer- Basic Design Issues, Error Correction & Detection Algorithms, Elementary Data Link Layer Protocols, Sliding Window Protocols, Error Handling, High Level Data Link Control 9

UNIT-III

Network Layer: Packet Switched Networks – IP – ARP – RARP –DHCP – ICMP – Queuing Discipline – Routing Algorithms, Congestion Control Algorithms, Internetworking, TCP/IP Protocol, IP Addresses, Ipv4 and Ipv6. 9

UNIT-IV

Transport Layer: Design Issues, Connection Management, Internet Transport Protocol (UDP), Transmission Control Protocol. (TCP) -**Adaptive Retransmission Congestion Control** 9

Congestion Avoidance – QOS.

Application Layer: Domain Name System, Electronic Mail **(Email),** File Transfer Protocol, Hyper Text Transfer Protocol, Introduction To Cryptography and Network Security.

EXPERIMENTS

1. To create scenario and study the performance of CSMA/CD protocol through simulation.
2. To create scenario and study the performance of token bus and token ring protocols through simulation.
3. Implementation of Error detection and correction algorithms.
4. Implementation and study of 1-bit sliding window viz., stop and wait protocol.
5. Implementation and study of Go back-N protocol.
6. Implementation and study of selective repeat protocol.
7. To get the MAC or Physical address of the system using Address Resolution Protocol.
8. Implementation of distance vector routing algorithm.
9. Implementation of link state routing algorithm.
10. To write a client-server application for chat using TCP.
11. To write a C program to develop a DNS client server to resolve the given hostname.

Textbooks

1. Forouzan, Data Communication and Networking, TMH
2. A. S Tanenbaum, Computer Networks, 4th , Edition", Pearson education

Reference books

1. W. Stallings , Data and Computer Communication , Macmillan Press
2. Comer , Computer Networks & Internet with Internet Applications, Pearson Education
3. Comer, Internetworking with TCP/IP, 6th Edition, PHI
4. W Stallings, Computer Networks with Internet Protocols, Pearson Education
5. W Stallings, Local and Metropolitan Area Networks, 6th edition, Pearson Education

BCS-35 SOFTWARE ENGINEERING

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Enhance the Software Project Management skills.
2. Develop functioning software which benchmarks to the international standards.

Topics Covered

UNIT-I

Software Process– Introduction, S/W Engineering Paradigm , Life Cycle Models (Waterfall, Incremental, Spiral, Evolutionary, Prototyping), Software Requirements –Functional And Non-Functional–Software Document– Requirement Engineering Process–Feasibility Studies – Software Prototyping–Prototyping in Software, Process–Data–Functional and Behavioral Models– Structured Analysis And Data Dictionary. 9

UNIT-II

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 Graph. 9

UNIT-III

Software Testing – Taxonomy of S/W Testing Levels - Black Box Testing – Testing Boundary Conditions – Structural Testing — Regression Testing– S/W Testing Strategies, Unit Testing, Integration Testing, Validation Testing, System Testing and Debugging. 9

UNIT-IV

Measures and Measurements – Zipf's Law, Software Cost Estimation – Function Point Models, COCOMO Model. Delphi Method – Scheduling – Earned Value Analysis – Error Tracking – Software Configuration Management – Program Evolution Dynamics – Software Maintenance – Project Planning – Project Scheduling– Risk Management – Case Tools 9

EXPERIMENTS

1. Identifying the Requirements from Problem Statements
2. Estimation of Project Metrics
3. Modeling UML Use Case Diagrams and Capturing Use Case Scenarios
4. E-R Modeling from the Problem Statements
5. Identifying Domain Classes from the Problem Statements
6. State chart and Activity Modeling
7. Modeling UML Class Diagrams and Sequence diagrams
8. Modeling Data Flow Diagrams
9. Estimation of Test Coverage Metrics and Structural Complexity
10. Designing Test Suites

Textbooks

1. R. S. Pressman, "Software Engineering - A practitioners approach", 3rd Edition, McGraw Hill International editions, 1992.

Reference books

1. IAN Sommerville, Software Engineering, Pearson Education Asia, VI Edition, 2000.

2. Pankaj Jalote, "An Integrated Approach to software Engineering", Springer Verlag, 1997

BCS-36 DATABASE MANAGEMENT SYSTEM, DATA MINING & WAREHOUSING

Course Category	:	For Other Department
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. To educate students with fundamental concepts of Database Management System, Data Models, Different Data Base Languages.
2. To analyze Database design methodology.
3. To understand the basic principles, concepts and applications of data warehousing and data mining
4. To introduce the task of data mining as an important phase of knowledge recovery process
5. Ability to do Conceptual, Logical, and Physical design of Data Warehouses OLAP applications and OLAP deployment
6. Have a good knowledge of the fundamental concepts that provide the foundation of data mining

Topics Covered

UNIT-I

Introduction: An Overview of Database Management System, 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. 9

Data Modeling using 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.

UNIT-II

Relational Data Model and Language: Relational Data Model Concepts, Integrity Constraints, 9

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

UNIT-III

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. 9

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.

UNIT-IV

Data Mining & Warehousing: Introduction to Data Warehouse, Building A Data Warehouse, Data Warehouse Architecture, OLAP Technology, Introduction to Data Mining, Data Pre-Processing, Mining Association Rules, Classification and Prediction, Cluster Analysis, Advanced Techniques of Data Mining and its Applications. 9

Textbooks

1. Korth, Silbertz, Sudarshan, Database Concepts, McGraw Hill
2. Jiawei Han, Micheline Kamber, Data Mining Concepts & Techniques, Elsevier
3. Elmasri, Navathe, Fundamentals of Database Systems, Addison Wesley

Reference books

1. Date C J, An Introduction to Database Systems, Addison Wesley
2. J. D. Ulman, Principles of Database and Knowledge base System, Computer Science Press.
3. M. H. Dunham, Data Mining: Introductory and Advanced Topics. Pearson Education
4. Mallach, Data Warehousing System, McGraw –Hill

BCS-37 NETWORK SECURITY & CRYPTOGRAPHY

Course Category	:	For Other Department
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this

course

1. Understand the basic concept of Cryptography and Network Security, their mathematical models
2. Various types ciphers, DES, AES, message Authentication, digital Signature, System
3. Network security, Viruses, worms and firewall
4. Understand mathematical foundation required for various cryptographic Algorithms.
5. DES, AES, IDEA and RC5 cryptographic technique
6. Public and Private Key cryptography.
7. Various Message Digest Algorithm,
8. Comprehend and apply email security services and mechanisms
9. Comprehend and apply IP security mechanisms
10. Comprehend and apply authentication services and mechanisms
11. Comprehend and apply WEB security mechanisms
12. Design of Firewall, Intrusion and Filtering

Topics Covered

UNIT-I

Introduction to Cryptography 9

Need, Attacks, Security Principles, Security Services, Conventional & Classical Encryption Techniques, Modern Techniques: Simplified DES, Block Cipher Principles, DES Standard, DES Strength, Differential & Linear Cryptanalysis, Block Cipher Design Principles, Block Cipher Modes of Operation

UNIT-II

Conventional Encryption Algorithm and Public Key Encryption 9

Triple DES, IDEA, RC5, AES, Key Distribution, Public Key Cryptography: Principles of Public Key Cryptosystem, RSA Algorithm, Key Management, Fermat's and Euler's Theorem, Chinese Remainder Theorem

UNIT-III

Hash Functions 9

Message Authentication and Hash Function: Authentication Requirements, Authentication Functions, Message Authentication Codes, Birthday Attacks, MD5 Message Digest Algorithm, Secure Hash Algorithm (SHA), Digital Signature, Authentication Protocol, Digital Signature Standard (DSS)

UNIT-IV

Network and System Security: 9

Authentication Applications: Kerberos, Electronic Mail Security, Pretty Good Privacy (PGP), S/MIME,IP Security: Authentication Header, Encapsulation Security Payload, Combining Security Association, Key Management, Web Security: Secure Socket Layer and Transport Layer security, Secure Electronic Transaction (SET), System security: Intruders, Viruses,

Worms, Firewall design principles

EXPERIMENTS

1. Implementation of DES Algorithm.
2. Implementation of Random number generation.
3. Implementation of AES Cryptographic technique.
4. Implementation of IDEA Cryptographic technique.
5. Implementation of RSA Algorithm.
6. Generate the Digital signature.
7. Implementation of MD5 Algorithm.
8. Implementation of SHA Algorithm.
9. Implementation of MD5 Algorithm.
10. Demonstrate and implement the PGP Algorithm.
11. Demonstrate and simulate the working of Firewall.

Textbooks

1. William Stallings, Cryptography and Network Security Principles and Practices, Sixth Edition, PHI Publication
2. Atul Kahate, Cryptography and Network Security, Second Edition, TMH Publication
3. Shyamla, Harini and Padmnabhan, Cryptography and Security, Wiley Publication
4. Deven Shah, Information Security Principles and Practice, Wiley-India
5. Forouzan, Mukhopadhyay, Cryptography & Network Security, McGraw Hill

Reference books

1. Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley and Sons
2. Godbole, Information Systems Security, Wiley-India
3. Arto Salomaa, Public-Key Cryptography by, second edition, Springer, 1996
4. Goodrich and Tamassia, Introduction to Computer Security , Addison-Wesley Publication
5. Rubin, Geer and Ranum, Web Security Sourcebook: A Complete Guide to Web Security Threats and Solutions , Wiley Publication
6. Henk C.A. van Tilborg, An Introduction to Cryptology, Kluwer Academic Publishers
7. N. Doraswamy and Dan Harkins, IPSec- **The New Security Standard for the Internet, Intranets, and Virtual Private Networks,** Prentice Hall, USA

BCS-40 PROJECT PART- I

Course category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture : 0, Tutorial : 0 , Practical: 10
Number of Credits	:	5
Course Assessment methods	:	Continuous assessment through three viva voce/presentation, preliminary project report, effort and regularity and end semester presentation
Course Outcomes	:	The students are expected to be able to demonstrate the

following knowledge, skills and attitudes after completing this course

1. Learning of latest trends and technology in selected field of interest.
2. Apply the acquired knowledge to practical situations.
3. Develop self-interest to explore the selected technical field of interest in future.
4. Acquire presentation skills.
5. Develop better interpersonal communication skills and increase self confidence

BCS-41 INTRODUCTION TO MACHINE LEARNING

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. To explain theory underlying machine learning
2. To construct algorithms to learn linear and non-linear models
3. To implement data clustering algorithms
4. To construct algorithms to learn tree and rule-based models
5. To apply reinforcement learning techniques

Topics Covered

UNIT-I

FOUNDATIONS OF LEARNING- Components of Learning – Learning Models – Geometric Models – Probabilistic Models – Logic Models – Grouping and Grading – Learning Versus Design – Types of Learning – Supervised – Unsupervised – Reinforcement – Theory of Learning – Feasibility of Learning – Error and Noise – Training versus Testing – Theory of Generalization – Generalization Bound – Approximation- Generalization Tradeoff – Bias and Variance – Learning Curve 9

UNIT-II

LINEAR MODELS-Linear Classification–Univariate Linear Regression–Multivariate Linear Regression–Regularized Regression– Logistic Regression–Perceptron–Multilayer Neural Networks –Learning Neural Networks Structures – Support Vector Machines–Soft Margin SVM–Going Beyond Linearity – Generalization and Over Fitting – Regularization– Validation 9

UNIT-III

DISTANCE-BASED MODELS-Nearest Neighbour Models–K-Means–Clustering around Medoids–Silhouettes–Hierarchical Clustering–K-D Trees–Locality Sensitive Hashing–Non-Parametric Regression–Ensemble Learning–Bagging And Random Forests–Boosting–Meta Learning 9

UNIT-IV

TREE AND RULE MODELS- Decision Trees – Learning Decision Trees – Ranking and Probability Estimation Trees – Regression Trees – Clustering Trees – Learning Ordered Rule Lists – Learning Unordered Rule Lists – Descriptive Rule Learning – Association Rule Mining – First-Order Rule Learning 9

UNIT-V

REINFORCEMENT LEARNING-Passive Reinforcement Learning – Direct Utility Estimation – Adaptive Dynamic Programming – Temporal-Difference Learning – Active Reinforcement Learning – Exploration – Learning an Action-Utility Function – Generalization in Reinforcement Learning – Policy Search – Applications in Game Playing – Applications in Robot Control

EXPERIMENTS

1. A simple *linear regression* attempts to draw a straight line that will best minimize the residual sum of squares between the observations and the predictions in python program language
2. Linear Regression Logistic Regression in python program language
3. Decision Tree in python program language
4. SVMin python program language
5. Naive Bayesin python program language
6. KNNin python program language
7. K-Meansin python program language
8. Random Forestin python program language
9. Dimensionality Reduction Algorithms in python program language
10. Gradient Boost & Adaboost in python program language

Textbooks

1 Ethem Alpaydın -Introduction to Machine Learning Third Edition, MIT Press, 2004

Reference books

1. Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, Learning from Data, AML Book Publishers, 2012.
2. P. Flach, Machine Learning: The art and science of algorithms that make sense of data, Cambridge University Press, 2012.
3. K. P. Murphy, Machine Learning: A probabilistic perspective, MIT Press, 2012.
4. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2007.
5. D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012.
6. M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, MIT Press, 2012.

7. T. M. Mitchell, Machine Learning, McGraw Hill, 1997.
8. S. Russel and P. Norvig, Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall, 2009.

BCS-42 PARALLEL & DISTRIBUTED COMPUTING

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. understand and account for models, limitations, and fundamental concepts in the area of message passing and shared memory concurrency, and apply this understanding to example systems and algorithms
2. adapt, and design algorithms for execution in parallel and distributed settings, and analyze the algorithms for correctness, reliability, security, and performance

Topics Covered

UNIT-I

Parallel Computing, Parallel Computer Model, Parallel Architectural Classification Schemes, Multiprocessor System and Interconnection Networks. Theoretical Foundation For Distributed System: Limitation of Distributed System, Absence of Global Clock, Shared Memory, Logical Clocks, Lamport's & Vectors Logical Clocks, Causal Ordering of Messages, Global State, Termination Detection. 9

UNIT-II

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. 9

UNIT-III

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. 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. 9

UNIT-IV

Distributed File Systems: File Service Architecture, Sun Network File System, The Andrew File System, Recent Advances. 9

Distributed Algorithms: Introduction to Communication Protocols, Balanced Sliding Window Protocol, Routing Algorithms, Destination Based Routing, APP Problem, Deadlock Free Packet Switching, Introduction to Wave & Traversal Algorithms, Election Algorithm, CORBA Case Study: CORBA RMI, CORBA Services.

EXPERIMENTS

1. Write a program to simulate the functioning of Lamport's logical clock in 'C'.
2. Write a program to simulate the Distributed Mutual Exclusion in 'C'.
3. Write a program to implement a Distributed chat server using TCP sockets in 'C'.
4. Implement RPC mechanism for a file transfer across a network in 'C'.
5. Write a JAVA code to implement 'JAVA RMI' mechanism for accessing methods of remote systems.
6. Write a code in 'C' to implement sliding window protocol.
7. Implement corba mechanism by using c++ program at one end and JAVA program at the other.
8. Write a code in 'C' to Increment a counter in shared memory.

Textbooks

1. Singhal Mukesh & Shivaratri N. G., Advanced Concepts in Operating Systems, TMH

Reference books

2. D. Culler, J. P. Singh, A. Gupta, Parallel Computer Architecture, Elsevier
3. Andrew S. Tanenbaum and Maarten Van Steen, Distributed Systems Principles and Paradigms, PHI
4. Tanenbaum, A. S. Distributed Operating Systems, Prentice Hall 199
5. Tanenbaum, A. S. Modern Operating Systems, 2nd Edition, Prentice Hall 2001.
6. Bacon, J., Concurrent Systems, 2nd Edition, Addison Wesley 1998.
7. Silberschatz, A., Galvin, P. and Gagne, G., Applied Operating Systems Concepts, 1st Edition, Wiley 2000.
8. Coulouris, G. et al, Distributed Systems: Concepts and Design, 3rd Edition, Addison Wesley 2001.
9. Galli, D.L., Distributed Operating Systems: Concepts and Practice, Prentice-Hall 2000.

BCS-43 MOBILE COMPUTING

Course Category	:	Department Core (DC)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 2
Number of Credits	:	5
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical

Examination

Course Outcomes : The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Demonstrate the actual meaning of power and energy management in wireless mobile networks.
2. Outline knowledge on Mobile IP.
3. Be familiar with the network protocol stack
4. Learn the basics of mobile telecommunication system
5. Be exposed to Ad-Hoc networks
6. Gain knowledge about different mobile platforms and application development

Topics Covered

UNIT-I

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. 9

UNIT-II

Wireless Networking, Wireless LAN Overview: MAC Issues, IEEE 802.11, Blue Tooth, Wireless Multiple Access Protocols, TCP Over Wireless, Wireless Applications, Mobile IP, WAP: Architecture, Protocol Stack, Application Environment, Applications, Wireless mark Up Language (WML). 9

UNIT-III

Data Management Issues, Data Replication for Mobile Computers, Adaptive Clustering for Mobile Wireless Networks, File System, Disconnected Operations, Mobile Agents Computing, Security and Fault Tolerance. 9

UNIT-IV

Adhoc 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 Network. 9

EXPERIMENTS

1. Develop an application that uses GUI components, Font and Colours
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Write an application that draws basic graphical primitives on the screen.
5. Implement an application that implements Multi threading
6. Develop a native application that uses GPS location information.
7. Implement an application that writes data to the SD card.
8. Implement an application that creates an alert upon receiving a message.

9. Write a mobile application that creates alarm clock

Textbooks

1. Asoke K Taukder, Roopa R Yavagal, Mobile Computing, Tata McGraw Hill Pub. Co., New Delhi, 2005.
2. J. Schiller, Mobile Communication, Addison Wesley, 2000.

Reference books

1. Ivan Stojmenovic, Handbook of Wireless Networks and Mobile Computing, John Wiley & sons Inc, Canada, 2002.
2. William Stallings, "Wireless Communication and Networks", Pearson Education, 2003.
3. Yi-Bing Lin & Imrich Chlamtac, Wireless and Mobile Networks Architectures, John Wiley & Sons, 2001.
4. Raj Pandya, "Mobile and Personal Communication systems and services", Prentice Hall of India, 2001.
5. Hansmann, "Principles of Mobile Computing", Wiley Dreamtech, 2004.
6. Ray Rischpater, "Wireless Web Development", Springer Publishing, 2000.
7. P. Stavronlakis, "Third Generation Mobile Telecommunication systems", Springer Publishers, 2001.
8. Burkhardt , Pervasive Computing, Pearson
9. P. Stavronlakis, Third Generation Mobile Telecommunication systems, Springer Publishers.

BCS-44 OBJECT ORIENTED TECHNIQUES & JAVA PROGRAMMING

Course Category	:	Program Elective (EC) Electrical Engineering
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Knowledge of how to develop and deploy applications and applets in JAVA.
2. Knowledge of how to develop and deploy GUI using JAVA Swing and AWT.
3. Design, develop and implement interactive web applications.
4. Be able to implement, compile, test and run JAVA programs comprising more than one class and to address a particular software problem.
5. Develop programs using the JAVA Collection API as well as the JAVA standard class

library.

Topics Covered

UNIT-I

Introduction: Introduction to Programming Languages, The Evolution of JAVA, Object-Oriented Programming Concepts and JAVA, Differences between C++ and JAVA, Primary Characteristics of JAVA, The Architecture, Programming with JAVA, Operator, Data type, Variable, Arrays, Control Statements, Methods. 9

UNIT-II

Core JAVA: Classes, Inheritance, Package and Interface, Exception Handling, Multithread Programming, I/O, JAVA Applet, String Handling, Networking, Event Handling, Introduction to AWT, AWT Controls, Layout Managers. 9

UNIT-III

JAVA Swing: Creating a Swing Applet and Application, Programming using Panes, Pluggable Look and Feel, Labels, Text Fields, Buttons, Tabbed Panes. 9

UNIT-IV

JDBC: Connectivity Model, JDBC/ODBC Bridge, JAVA.SQL Package, Connectivity to Remote Database, **JAVA Beans**: Application Builder Tools, The Bean Developer Kit(BDK), JAR files, Introspection, Developing a Simple Bean, **Servlet:** Introduction to JAVA Servlet: Servlet Basics, Servlet API Basic, Life Cycle of a Servlet, Running Servlet. 9

Textbooks

1. Naughton, Schildt, The Complete Reference JAVA2, TMH Publication
2. Balaguruswamy E, Programming in JAVA, TMH Publication

Reference books

1. Margaret Levine Young, The Complete Reference Internet, TMH Publication
2. Dustin R. Callway, Inside Servlets, Addison Wesley.
3. Mark Wutica, JAVA Enterprise Edition, QUE.
4. Steven Holzner, JAVA2 Black book, Dreamtech.

BCS-45 INDUSTRIAL / PRACTICAL TRAINING

Course category	:	Audit Course (AC)
Pre-requisite Subject	:	NIL
Contact hours/week	:	Lecture : 0, Tutorial : 0 , Practical: 2
Number of Credits	:	1
Course Assessment methods	:	Continuous assessment through technical quality of the work, attendance, discipline, involvement and interest, project work, viva voce, project report and presentation
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. The main objective of the Industrial Training is to experience and understand real life situations in industrial organizations and their related environments and accelerating the learning process of how student's knowledge could be used in a realistic way.
2. In addition to that, industrial training also makes one understand the formal and informal relationships in an industrial organization so as to promote favourable human relations and teamwork. Besides, it provides the exposure to practice and apply the acquired knowledge "hands - on" in the working environment.
3. Industrial training also provides a systematic introduction to the ways of industry and developing talent and attitudes, so that one can understand how Human Resource Development works. Moreover, students can gain hands-on experience that is related to the student understanding so that the student can relate to and widen the skills that have been learnt while being in university. Industrial training also exposes the students to the real career world and accustoms them to an organizational structure, business operation and administrative functions.
4. Furthermore, students implement what they have learned and learn more throughout this training. Besides, students can also gain experience to select the optimal solution in handling a situation. During industrial training students can learn the accepted safety practices in the industry.
5. Students can also develop a sense of responsibility towards society

BCS-50 PROJECT PART-II

Course category	:	Department Core (DC)
Pre-requisite Subject	:	Project Part-I (BCS-40)
Contact hours/week	:	Lecture : 0, Tutorial : 0 , Practical: 10
Number of Credits	:	5
Course Assessment methods	:	Continuous assessment through three viva voce/presentation, final project report, contribution made to literary world and Major examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. B. Tech. project is designed to allow students to work with faculty members on *one long project* that may require effort over two semesters. The final year project gives students an excellent opportunity to develop and demonstrate their innovation skills, design skills and research interests. These projects quite often lead to publications of their original work.

2. Develops ability of report writing.
3. Develops ability to be aware of current trends in specific area of interest

BCS-51 ADVANCE JAVA

Course Category	:	Program Elective (PE1&PE2)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Be proficient in using JAVA Servlets and related Web development tools
2. Identify different components of client/server Architecture on Internet computing
3. Design, develop and implement interactive Web applications
4. Know how to develop and deploy applications and applets in JAVA
5. Know how to design and develop GUI using JAVA Swing and AWT

Topics Covered

UNIT-I

Collections: Collection Interfaces, Concrete Collections, Collections Framework. 9

Multithreading : Creating Thread and Running it, Multiple Thread Acting on Single Object, Synchronization, Thread Communication, Thread Group, Thread Priorities, Daemon Thread, Life Cycle of Thread.

UNIT-II

Networking: Internet Addressing, Internet address, Factory Methods, Instance Methods, TCP/IP Client Sockets, URL, URL Connection, TCP/IP Server Sockets, Datagrams 9

Enterprise JAVA Bean: Preparing a Class to be a JAVA bean, Creating a JAVA bean, JAVA bean Properties, Types of Beans, Stateful Session Bean, Stateless Session Bean, Entity Bean

UNIT-III

JAVA Database Connectivity (JDBC): 9

Merging Data from Multiple Tables: Joining, Manipulating Databases with JDBC, Prepared Statements, Transaction Processing, Stored Procedures C

Servlets: Servlet Overview and Architecture, Interface Servlet and Servlet Life Cycle, Handling HTTP Get Requests, Handling HTTP Post Requests, Redirecting Requests to Other Resources, Session Tracking, Cookies, Session Tracking with Http session.

UNIT-IV

JAVA Server Pages (JSP): Introduction, JAVA server Pages Overview, A First JAVA server Page Example, Implicit Objects, Scripting, Standard Actions, Directives, Custom Tag Libraries 9

Remote Method Invocation: Defining Remote Interface, Implementing Remote Interface, Compiling and Executing Server and Client.

Common Object Request Broker Architecture (CORBA): Technical/Architectural Overview, CORBA Basics, CORBA Services

Introduction Smart Phone Application Development: Introduction to Android Platform, Creating Application Template, Adding Activity, Intent, Services to Application, Using Google Map API

Textbooks

1. H. M. Deitel, P. J. Deitel, S. E. Santry , Advanced JAVA 2 Platform HOW TO PROGRAM, Prentice Hall

Reference books

2. Antonio Goncalves, Beginning JAVA™ EE 6 Platform with Glass Fish 3 From Novice to Professional.

BCS-52 .NET TECHNOLOGY

Course Category	:	Program Elective (PE1&PE2)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous Assessment through Tutorials, Assignments, Quizzes and Three Minor Tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand the most important features of .NET Framework technology
2. Use Visual Studio .NET and .NET Framework SDK to design, run and debug simple C# console applications
3. Write programs that use fundamental C# programming tools.
4. Use advanced OOP tools when designing C# programs.
5. Design web forms using ASP.Net

Topics Covered

UNIT-I

The .Net Framework: Introduction, Origin of .Net Technology, Common Language Runtime (CLR), Common Type System (CTS), Common Language Specification (CLS), Microsoft Intermediate Language (MSIL), Just-in –Time Compilation, Framework Base Classes. 9

C -Sharp Language (C#): Introduction, Data Types, Identifiers, Variables, Constants, Literals, Array And Strings, Object And Classes.

UNIT-II

C -Sharp Language (C#) (Cont.): Inheritance and Polymorphism, Operator Overloading, Interfaces, Delegates and Events. Type Conversion. 9

C# Using Libraries: Namespace- System, Input-Output, Multi-Threading

UNIT-III

Managing Console I/O Operations, Windows Forms, Error Handling. 9

Advanced Features Using C#: Web Services, Window Services, Unsafe Mode, Graphical Device Interface With C#, Introduction About Generic.

UNIT-IV

ASP .Net: Web Forms in ASP.NET, States, Validation, Login, ASP.NET Administrative Tasks, Learning about SQL Basics and Advanced Queries, ADO.NET, ASP.NET Data Controls, Ajax Extensions, LINQ, Working With XML Data, Web Services. 9

Textbooks

1. Deitel et al. Visual C# 2012 How to program. Prentice-Hall Inc., 2014, Fifth Edition
2. Aitken, Peter G. .NET Graphics and Printing Optimax Publishing, 2003
3. Prosise, Jeff. Programming Microsoft .NET Microsoft Press, 2002

Reference books

1. Wrox, Beginning Visual C# 2008,Wiley
2. Fergal Grimes, Microsoft .Net for Programmers. SPI
3. Balaguruswamy, Programming with C#, TMH
4. Mark Michaelis, Essential C# 3.0: For .NET Framework 3.5, 2/e, Pearson Education
5. Shibi Parikkar, C# with .Net Frame Work, Firewall Media
6. Wrox, Beginning ASP.NET 4.5 in C# and VB, 2012
7. Lippman, Stanley B. C# Primer - A Practical Approach Addison-Wesley, 2012

BCS-53 LAMP TECHNOLOGY

Course Category	:	Program Elective (PE1&PE2)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this

course.

1. Use Open Source Operating system and its distributions like Fedora, Google chrome OS, Ubuntu.
2. To comprehend framework of BSD (Berkley System Distribution) and its installation
3. Study of Web technologies based on open Software's LAMP (Linux Apache MySql and PHP/Python)
4. To Learn HTML, XHTML, PHP and JAVA Script

Topics Covered

UNIT-I

Introduction to LAMP Terminologies, Two Tier and Three Tier Web based Application Architecture; Advantages of using LAMP based Technologies, Linux: Distributions – Fedora and Ubuntu; Installation – Disk Partitioning, Boot Loader, Etc; Using Linux – Shell, File System Familiarity; Linux Administration – Managing Users, Services and Software; Network Connectivity and Configurations; Security. 9

UNIT-II

Apache: Web Server Conceptual Working, Web Browser, HTTP, Installation and Configuration; *Httpd. Conf* File; Logging; Security; Running Website 9

UNIT-III

Mysql: Database Management System, ER Diagram, Relational Database, Installation, Configuration, Administration, Common SQL Queries – Create, Describe, Select, Insert, Delete, Update, Etc. 9

UNIT-IV

PHP: Dynamic Content, Server Side Scripting, Installation, Configuration, Administration, Language Syntax, Built-in Functions, PHP and Mysql Connectivity, Installation, Configuration and Administration of All Four LAMP Components Namely Linux, Apache Web Server, Mysql and PHP, Testing with Any Project Example. 9

Textbooks

1. Eric Rosebrock, Setting Up LAMP, Sybex Publishers.
2. James Lee, Brent Ware, Open Source Development with LAMP, Addison-Wesley Professional.
3. Jason Gerner, Elizabeth Naramore, Professional LAMP, John Wiley & Sons.

Reference books

1. Ben Laurie, Peter Laurie, Apache – Definitive Guide, O'Reilly Publications.
2. Paul DuBois, MySQL, Addison-Wesley.
3. Rasmus L Erdorf, Kevin Tatroe, Programming PHP, O'Reilly Publications.

BCS-54 NETWORK PROGRAMMING

Course Category : Program Elective (PE1&PE2)
Pre-requisite Subject : NIL

Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. To write socket API based programs
2. To design and implement client-server applications using TCP and UDP sockets
3. To analyze network programs

Topics Covered

UNIT-I

Introduction, Overview of UNIX OS, Environment of a UNIX Process, Process Control, Process Relationships, Signals, Inter-process Communication, Overview of TCP/IP Protocols. 9

UNIT-II

Elementary TCP Sockets- Introduction to Socket Programming: Introduction to Sockets, Socket Address Structures, Byte Ordering Functions, Address Conversion Functions, Elementary TCP Sockets, socket, connect, bind, listen, accept, read, write, close functions, Iterative Server, Concurrent Server. 9

UNIT-III

TCP Echo Server, TCP Echo Client, Posix Signal Handling, Server with Multiple Clients, Boundary Conditions: Server Process Crashes, Server Host Crashes, Server Crashes and Reboots, 9

Server Shutdown, I/O Multiplexing, I/O Models, Select Function, Shutdown Function, TCP Echo Server (with Multiplexing), Poll Function, TCP Echo Client (with Multiplexing).

UNIT-IV

Socket Options, Getsocket and Setsocket Functions, Generic Socket Options, IP Socket options, ICMP Socket Options ,TCP Socket Options, Elementary UDP Sockets, UDP Echo Server, UDP Echo Client, Multiplexing TCP and UDP Sockets, Domain Name System, Gethos by name Function, Ipv6 Support in DNS, Gethostbyadr Function, Getserv by name and Getserv by port Functions. 9

Textbooks

1. W. Richard Stevens, S.A Rago, Programming in the Unix environment, 2nd edition, Pearson, 2005.

Reference books

1. W. Richard Stevens, B. Fenner, A.M. Rudoff, Unix Network Programming – The Sockets Networking API, 3rd edition, Pearson, 2004.
2. W. Richard Stevens, TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994

BCS-55 MOBILE APPLICATION PROGRAMMING

Course Category	:	Program Elective (PE1&PE2)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Know the components and structure of mobile application development frameworks for Android and windows OS based mobiles.
2. Understand how to work with various mobile application development frameworks.
3. Learn the basic and important design concepts and issues of development of mobile applications.
4. Understand the capabilities and limitations of mobile devices.

Topics Covered

UNIT-I

Android Development, Android Operation System, Important Android Components, Security and Permissions, Android SDK, Eclipse, Create an Android Emulator Device, Design, Develop and Deploy Application on a Real Device. 9

UNIT-II

Introduction to Windows Phone 7, Type of applications that can be built- using Silverlight and XNA, Developer tools to be used for building apps 9

UNIT-III

Introduction to App Makr, Creating a Developer Account on App Hub: using a Dream Spark Account, App Certification Guidelines for the Windows Phone Marketplace 9

UNIT-IV

iOS overview, iOS Application Life Cycle, Design, Develop and Deploy Applications for iPhone, iPad and iPod Touch, Human Interface and use of Sensors for App Development. 9

Textbooks

1. Jeff Mcwherter, Scott Gowell, Professional Mobile Application Development, Wrox Publisher (2012), 1e

Reference books

1. Lauren Darcy, Shane Conder, Sams Teach Yourself Android Application Development in 24 Hrs, 1e

2. Himanshu Dwivedi, Chris Clark, David Thiel, Mobile Application Security, Tata McGraw Hill (2010), 1e

BCS-56 LINUX ADMINISTRATION AND SYSTEM CALL PROGRAMMING

Course Category	:	Program Elective (PE1&PE2)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. use the LINUX based system through various commands
2. understand the task of LINUX system administration
3. write programs for system programming like IPC, semaphore etc.

Topics Covered

UNIT-I

History of Unix and Linux, Architecture of Linux, Advantages of Linux, Introduction to Kernel, Introduction to Linux Shell: Types of Shell, Feature and Benefits of Shell. I/O Redirection and Piping, Pipes, Filters, Introduction to Various Text Editor, Various Vi Editing Modes, Scrolling, Yank and Paste, Put and Delete, Set Commands, Comparison of Emacs Editor, Vi Editor, Pico Editor. 9

UNIT-II

Introduction to Linux Files: Rules for Creating Files, Linux Files System, File Printing, Searching Files using Grep, Change Permission to Set Files and Change Owner of Files. Process, Listening with Ps, Killing with Kill, PID, UID, GID, Signals, Nice, Renice. 9

UNIT-III

General Administration Issues: Root Account, Creating User in Linux, Changing Password, Deleting User, Disabling User Account, Linux Password & Shadow File Formats System Shutdown and Restart Creating Groups, Custom Configuration and Administration Issues, Simple Commands 9

UNIT-IV

System Call Programming: System Calls, Usage of File Related System Calls through C Programming. Process: Concept, Types, Related Commands & System Calls, Usage of Process Related System Calls through C Programming 9

Textbooks

1. Ellen Siever, Robert Love and Arnold Robbins, Linux in Nutshell, Fifth Edition, Oreilly Media.
2. Kurt Wall, Mark Watson, Mark Whitis, Linux Programming, Third Edition, SAMS Techmedia
3. Mark Sobell, Practical Guide to Linux Programming, Pearson Education.

Reference books

1. Graham Glass & King Ables, UNIX for Programmers and Users , Pearson Education
2. J.Purcell, Linux Complete Command Reference, Red Hat Software, McGraw hill.

BCS-57 DATABASE ADMINISTRATION WITH ORACLE

Course Category	:	Program Elective (PE1&PE2)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous Assessment through Tutorials, Assignments, Quizzes and Three Minor Tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Gain a conceptual understanding of the Oracle database architecture and how its components work and interact with one another.
2. will also learn how to create an operational database and properly manage the various structures in an effective and efficient manner including performance monitoring, database security, user management, and backup/recovery techniques
3. Establish and in depth understanding of Database Administration using the DBMS Interfaces
4. Create and understand the application of user rolls, privileges, and the security of the database.
5. Discuss and understand the concepts of Backup and Recovery Procedures

Topics Covered

UNIT-I

Introduction: DBMS Architecture and Data Independence, DBA Roles and Responsibilities, SQL *PLUS Overview: SQL Plus Fundamentals, Producing More Readable Outputs, Accepting Values At Urntime, Using Isql *Plus, Modifying Data: Introduction to DML Statements, Truncating A Table, Transaction Control Language, Managing Constraints: Creating Constraints, Dropping Constraints, Enabling and Disabling Constraints, Deferring Constraints Checks 9

UNIT-II

Managing Views: Creating and Modifying Views, Using Views, Inserting, Updating and Deleting 9

Data through Views, User Access and Security: Creating and Modifying Use Accounts, Creating and Using Roles, Granting and Revoking Privileges, Managing User Groups with Profiles, Oracle Overview and Architecture: An Overview of Logical and Physical Storage Structures, Oracle Memory Structures, Oracle Background Processes, Connecting to Oracle Instance, Processing SQL Command., Managing Oracle: Starting Up the Oracle Instance, Managing Sessions, Shutting Down the Oracle Instance, Instances Messages and Instance Alerts.

UNIT-III

Control and Redo Log Files: Managing the Control Files, Maintaining and Monitoring Redo Log Files, Managing Tables, Indexes and Constraints: Storing Data (Create, Alter, Analyzing, Querying Table Information), Managing Indexes, Managing Constraints, Managing Users and Security: Profiles, Managing Users, Managing Privileges, Managing Roles, Querying Role Information, Introduction to Network Administration: Network Design Considerations, Network Responsibilities for the DBA, Network Configuration, Overview of Oracle Net Features, Oracle Net Stack Architecture 9

UNIT-IV

Backup and Recovery Overview: Database Backup, Restoration and Recovery, Types of Failure in Oracle Environment, Defining A Backup and Recovery Strategy, Testing the Backup and Recovery Plan, Introduction to Performance Tuning: Brief Overview of Tuning Methodology, General Tuning Concepts 9

Textbooks

1. C.J. Date, Database Systems, Addison Wesley, 2000
2. Chip Dawes, Biju Thomas, Introduction to Oracle 9i SQL, BPB, 2002
3. Bob Bryla, Biju Thomas, Oracle 9i DBA Fundamental I, BPB, 2002
4. Doug Stums, Matthew Weshan, Oracle 9i DBA Fundamental I, BPB, 2002
5. Joseph C. Johnson, Oracle 9i Performance Tuning., BPB, 2002

Reference books

1. Loney and Koch, Oracle Database 10g: The Complete Reference, McGraw-Hill Osborne Media
2. Joseph C. Johnson, Oracle 9i Performance Tuning., BPB, 2002

BCS-58 DATA WAREHOUSING & DATA MINING

Course Category	:	Program Elective (PE1&PE2)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Approach business problems data-analytically by identifying opportunities to derive business value from data.
2. know the basics of data mining techniques and how they can be applied to extract relevant business intelligence

Topics Covered

UNIT-I

Introduction to Data Mining: Motivation for Data Mining, Data Mining-Definition &Functionalities, Classification of DM Systems, DM Task Primitives, Integration of a Data Mining System with A Database or A Data Warehouse, Major Issues in Data Mining. Data Warehousing .Overview of Concepts Like Star Schema, Fact and Dimension Tables, OLAP Operations, from OLAP to Data Mining. Data Pre Processing: Why? Descriptive Data Summarization, Data Cleaning: Missing Values, Noisy Data, Data Integration and Transformation. Data Reduction: Data Cube Aggregation, Dimensionality Reduction, Data Compression, Numerosity Reduction, Data Discretization and Concept Hierarchy Generation for Numerical and Categorical Data. 9

UNIT-II

Mining Frequent Patterns, Associations and Correlations: Market Basket Analysis, Frequent Item Sets, Closed Item Sets, and Association Rules, Frequent Pattern Mining, Efficient and Scalable Frequent Item Set Mining Methods, The Apriori Algorithm for Finding Frequent Item Sets Using Candidate Generation, Generating Association Rules from Frequent Item Sets, Improving the Efficiency of Apriori, Frequent Item sets without Candidate Generation using FP Tree, Mining Multilevel Association Rules, Mining Multidimensional Association Rules, from Association Mining to Correlation Analysis, Constraint-Based Association Mining. Issues Regarding Classification and Prediction: Classification Methods: Decision Tree, Bayesian Classification, Rule Based Prediction: Linear and Non Linear Regression Accuracy and Error Measures, Evaluating the Accuracy of A Classifier or Predictor. 9

UNIT-III

Cluster Analysis: Types of Data in Cluster Analysis, Categories of Clustering Methods, Partitioning Methods K-Means, K-Mediods Hierarchical Clustering-Agglomerative and Divisive Clustering, BIRCH and ROCK Methods, DBSCAN, Outlier Analysis Stream Data Classification, Clustering Association Mining in Stream Data. Mining Sequence Patterns in Transactional Databases 9

UNIT-IV

Spatial Data and Text Mining: Spatial Data Cube Construction and Spatial OLAP, Mining Spatial Association and Co-location Patterns, Spatial Clustering Methods, Spatial Classification and Spatial Trend Analysis. Text Data Analysis and Information Retrieval, Dimensionality Reduction for Text, Text Mining Approaches Web Mining Introduction, Web Content Mining, Web Structure Mining, Web Usage Mining, Automatic Classification of Web Documents. Data Mining for Business Applications like Balanced Scorecard, Fraud Detection, Click Stream Mining, Market Segmentation, Retail Industry, Telecommunications Industry, Banking & Finance and CRM etc. 9

Textbooks

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 2nd Edition
2. P. N. Tan, M. Steinbach, Vipin Kumar, .Introduction to Data Mining., Pearson Education

Reference books

1. MacLennan Jamie, Tang Zhao Hui and Crivat Bogdan, .Data Mining with Microsoft SQL Server 2008, Wiley India Edition.
2. G. Shmueli, N.R. Patel, P.C. Bruce, .Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XL Miner, Wiley India.
3. Michael Berry and Gordon Linoff .Data Mining Techniques., 2nd Edition Wiley Publications
4. Alex Berson and Smith, .Data Mining and Data Warehousing and OLAP, McGraw Hill Publication.
5. E. G. Mallach, .Decision Support and Data Warehouse Systems", Tata McGraw Hill.
6. Michael Berry and Gordon Linoff .Mastering Data Mining- Art & science of CRM., Wiley Student Edition
7. Arijay Chaudhary & P. S. Deshpande, Multidimensional Data Analysis and Data Mining Dreamtech Press
8. Vikram Pudi & Radha Krishna, .Data Mining, Oxford Higher Education.

BCS-59 ANALYTICS AND SYSTEMS OF BIG DATA

Course Category	:	Program Elective (PE1&PE2)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Demonstrate the knowledge of big data, data science, data analytics, distributed file systems, parallel Map Reduce paradigm, NoSQL, machine learning, etc
2. Program and implement examples of big data and NoSQL applications using open source Hadoop, HDFS, Map Reduce, Hive, Pig, Mahout, etc
3. Read current research papers and implement example research group project in big data

Topics Covered

UNIT-I

Big Data, Complexity of Big Data, Big Data Processing Architectures, Big Data Technologies, Big Data Business Value, Data Warehouse, Re-Engineering the Data Warehouse, Workload Management in the Data Warehouse, New Technology Approaches. 9

Integration of Big Data and Data Warehouse, Data Driven Architecture, Information Management and Lifecycle, Big Data Analytics, Visualization and Data Scientist, Implementing the "Big Data" Data. Choices in Setting Up R for Business Analytics, R Interfaces, Manipulating Data, Exploring Data, Building Regression Models, Clustering and Data Segmentation, Forecasting and Time Series Models

UNIT-II

Writing Hadoop Map Reduce Programs, Integrating R and Hadoop, Using Hadoop Streaming with R, Learning Data Analytics with R and Hadoop, Understanding Big Data Analysis with Machine Learning. Big Data, Web Data, A Cross-Section of Big Data Sources and the Value They Hold, Taming Big Data, The Evolution of Analytic Scalability. 9

UNIT-III

The Evolution of Analytic Processes, The Evolution of Analytic, Processes the Evolution of Analytic Tools and Methods. Legacy Data, Hypothesis Testing, Prediction, Software, Complexity, Business problems suited to Big Data Analytics. 9

UNIT-IV

High Performance Appliances for Big Data Management using Graph Analytics, The New Information Management Paradigm, Big Data's Implication for Businesses, Big Data Implications for Information Management, Splunk's Basic Operations on Big Data. 9

Textbooks

1. Anand Rajaraman, Jure Leskovec, and Jeffrey D. Ullman, Mining of Massive Data Sets, Cambridge University Press. 2011.
2. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.

Reference books

1. Viktor Mayer Schönberger, Kenneth Cukier, Big Data: A Revolution That Will Transform How We Live, Work, and Think, John Murray 2013
2. Pramod J. Sadalage, Martin Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison Wesley
3. Eric Sammer, Hadoop Operation, O'Reilly 2012
4. Donald Miner, Adam Shook, MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems, O'Reilly 2012
5. "Big Data Now", by O'Reilly Media Inc., O'Reilly 2012

BCS-60 GAME THEORY

Course Category	:	Program Elective (PE1&PE2)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this

course.

1. Discuss the basics of games and the mathematics for games as well as the typical application areas for game theory.
2. Explain the concepts of non-cooperative and cooperative games and the basic computational issues.
3. Describe the concepts of Games with Perfect Information as well as Games with Imperfect Information.
4. Study the non-cooperative game theory.
5. Designing the mechanisms and understand the computational applications of mechanism design.

Topics Covered

UNIT-I

Introduction – Making Rational Choices: Basics of Games – Strategy - Preferences – Payoffs – Mathematical Basics -Game Theory-Rational Choice - Basic Solution Concepts-Non-Cooperative versus Cooperative Games - Basic Computational Issues - Finding Equilibrium and Learning in Games- Typical Application Areas for Game Theory (e.g. Google's Sponsored Search, ebay Auctions, Electricity Trading Markets) 9

UNIT-II

Games With Perfect Information- Strategic Games - Prisoner's Dilemma, Matching Pennies- Nash Equilibrium- Theory and Illustrations - Cournot's and Bertrand's Models of Oligopoly- Auctions- Mixed Strategy Equilibrium- Zero-Sum Games- Extensive Games with Perfect Information-Repeated Games (Prisoner's Dilemma)- Sub Game Perfect Nash Equilibrium; Computational Issues. 9

Games with Imperfect Information- Bayesian Games – Motivational Examples – General Definitions –Information Aspects – Illustrations - Extensive Games with Imperfect - Information - Strategies- Nash Equilibrium – Beliefs and Sequential Equilibrium – Illustrations - Repeated Games - Prisoner's Dilemma - Bargaining

UNIT-III

NON-COOPERATIVE GAME THEORY-Non-Cooperative Game Theory - Self-Interested Agents- Games in Normal Form – Analyzing Games: from Optimality to Equilibrium - Computing Solution Concepts of Normal-Form Games - Computing Nash Equilibrium of Two-Player, Zero-Sum Games -Computing Nash Equilibrium of Two-Player, General-Sum Games - Identifying Dominated Strategies 9

UNIT-IV

MECHANISM DESIGN-Aggregating Preferences-Social Choice – Formal Model- Voting - Existence of Social Functions - Ranking Systems- Protocols for Strategic Agents: Mechanism Design - Mechanism Design with Unrestricted Preferences- Efficient Mechanisms -Vickrey and VCG Mechanisms (Shortest Paths) - Combinatorial Auctions - Profit Maximization Computational Applications of Mechanism Design -Applications in Computer Science - Google's Sponsored Search - ebay Auctions 9

Textbooks

1. Kevin Leyton-Brown, Yoav Shoham, Ronald J Brachman, Thomas Dietterich, Essentials of Game Theory, Morgan and Claypool Publishers, 2008
2. Roger A McCain, Game Theory: A Nontechnical Introduction to the Analysis of Strategy,
3. Fudenberg, Drew, and Jean Tirole, Game Theory, Cambridge, MA: MIT Press, 1991
4. Osborne, Martin, and Ariel Rubinstein. A Course in Game Theory. Cambridge, MA: MIT Press, 1994
5. Mailath, George J., and Larry Samuelson, Repeated Games and Reputations. New York, NY: Oxford University Press, 2006
6. Weibull, Jorgen. Evolutionary Game Theory. Cambridge, MA: MIT Press, 1995.

Reference books

1. M. J. Osborne, An Introduction to Game Theory. Oxford University Press, 2004.
2. M. J. Osborne and A. Rubinstein, A Course in Game Theory. MIT Press, 1994.
3. N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani (Editors), Algorithmic Game Theory. Cambridge University Press, 2007
4. A. Dixit and S. Skeath, Games of Strategy, Second Edition. W W Norton & Co Inc, 2004.
5. Yoav Shoham, Kevin Leyton-Brown, Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Cambridge University Press 2008
6. Zhu Han, Dusit Niyato, Walid Saad, Tamer Basar and Are Hjorungnes, Game Theory in Wireless and Communication Networks, Cambridge University Press, 2012

BCS-66 ADVANCE PROGRAMMING TECHNIQUES

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Develop algorithms from user problem statements.
2. Express the solutions to computer oriented problems using pseudo code.
3. Proficiently transform designs of problem solutions into a standard programming language.
4. Use an integrated programming environment to write, compile, and execute programs involving a small number of source files.
5. Apply debugging and testing techniques to locate and resolve errors, and to determine

the effectiveness of a program.

6. Apply standard/structured programming techniques including design approaches, use of functions/methods, use of documentation, and avoidance of excessive branching.
7. Proficiently use fundamental programming elements including: variable declaration, use of data types and simple data structures (arrays and objects), decision structures, loop structures, input and output for console and text files, and functions/methods.

Topics Covered

UNIT-I

Introduction-History of Computers, Components of a Computer, Programming Languages, Compilation vs. Interpretation, Basic Program Structure and the Integrated Development Environment-Essential Program Structure, Documentation and Standard Programming Practices, Integrated Development Environment(IDE) Overview, Editing (with the IDE), Compilation (with the IDE), Execution (with the IDE), Debugging (with the IDE) 9

UNIT-II

Algorithm Development using Psuedo-code-Software Engineering Method, Procedural Problem Solving Approaches, Assignments, Conditionals, Loops, Classic Formula Problems, Classic Aggregate Problems (E.G., Maximum, Minimum, Sum, Average), 9

Basic Input And Output-Console Output including Basic Data Formatting, Console Input

Variables and Expressions-Variable Declarations including Common Data Types (E.G. Int, Float, String), Arithmetic, Expressions Including Precedence and Associativity, Assignment Statements (Numeric and String Data), Library Functions, Standard Programming Practices for Variables and Assignments, Case Problems Using Variables and Expressions

UNIT-III

Decision Structures-Boolean Expressions, Single Alternative Conditional Statements (E.G., If), Double Alternative Conditional Statements (E.G., If/Else), Multi-Way Statements (E.G., Case), Nested Conditional Structures, Standard/ Structures Programming Practices for Decision Structures, Case Problems using Decisions Structures 9

Loop Structures-Loop Control Variables, Initialization, Test and Modifications, Pre-Test Loop (E.G., While Loop), Post-Test Loop (E.G., Do-While Loop), Counting Loop (E.G., For Loop), Nested Loop Structures, Standard/ Structures Programming Practice for Loop Structures, Case Problems using Loop Structures

Input and Output using Files-Input Streams from Files, Priming Read Loop, Output Streams to Files, Case Problems using File Input and Output

UNIT-IV

Simple Data Structures-One Dimensional Arrays, Strings as Arrays, Multi-Dimensional Arrays, Records (E.G., Objects/Entities), Case Problems using Arrays and Records 9

Functions-Argument Passing, Returning Results, Recursion, Testing A Program System, Standard/Structures Programming Practices for Functions, Case Problems using Functions

Introduction to the Object Oriented Approach-Class Declarations, Instance Variables, Methods, Object Instantiation, Standard/Structures Programming Practice for Classes, Case Problems using Objects

Textbooks

1. Gaddis Tony, Starting Out with C++: From control structures through objects, 7th Edition, Addison-Wesley Publishing, 2012.

Reference books

1. Deitel & Deitel, JAVA: How to Program, 9th Edition, Prentice Hall, 2012.
2. Deitel & Deitel, C++ How to Program: Late Objects Version, 7th Edition, Prentice Hall, 2011.
3. Gaddis, Tony, Starting Out with JAVA: Control Structures to Objects, 2nd Edition, Pearson, 2012.
4. Horstmann, Cay, JAVA Concepts, 6th Edition, Wiley, 2009.
5. Liang, Y. Daniel, Introduction to Programming with JAVA, 8th Edition, Pearson, 2010.
6. Liang, Y. Daniel, Introduction to Programming with C++, 2nd Edition, Pearson, 2010.
7. Lewis, John, and Loftus, William, JAVA Software Solutions: Foundations of Program Design, 7th Edition, Pearson, 2012.
8. Malik, D. S., JAVA Programming: From Problem Analysis to Program Design, 5th Edition, Course Technology, 2011.
9. Malik, D. S., C++ Programming: From Problem Analysis to Program Design, 5th Edition, Course Technology, 2010.
10. Savitch, Walter, Absolute JAVA, 4th Edition, Addison Wesley, 2009.
11. Stroustrup, Bjarne, Programming: Principles and Practice Using C++, Addison-Wesley Professional, 2008.

BCS-67 COMPUTER VISION: FOUNDATIONS AND APPLICATIONS

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand the various operations performed on 2D image.
2. To recover the information, knowledge about the objects in the scene and projection geometry and understanding of 3D image.

Topics Covered

UNIT-I

Introduction: Computer Vision, Brief History. Image Formation: Geometric Primitives and Transformations, Photometric Image Formation, Digital Camera. Image Processing: Point Operators, Linear Filtering, Neighborhood Operators, Fourier Transform, Pyramids and Wavelet, 9

Geometric Transforms, Global Optimization.

UNIT-II

Feature Detection and Matching: Points and Patches, Edges, Lines. **Segmentation:** Active Contours: Snakes, Dynamic Snake and Condensation, Scissor, Level Sets, Split and Merge, Mean Shift and Mode Finding, **Feature Based Alignment:** 2D and 3D Feature Based Alignment, Pose Estimation, Geometric Intrinsic Calibration. 9

UNIT-III

Structure from Motion: Triangulation, Two Frame Structure from Motion, Factorization, Bundle Adjustment. **Dense Motion Estimation:** Translational Alignment, Parametric Motion, Spline Based Motion, Layered Motion. **Image Stitching:** Motion Models, Global Alignment, Composing. 9

UNIT-IV

3D Reconstruction: Surface Representation, Point based Representation, Volumetric Representation, Model based Reconstruction, Application: 3D Photography. **Image Based Rendering:** View Interpolation, Layered Depth Images, Video based Rendering. **Recognition:** Object Detection, Face Recognition, Context and Scene Understanding. 9

Textbooks

1. R. Szeliski, Computer Vision: Algorithms and Applications, Springer.
2. D. Forsyth and J. Ponce, Computer Vision- A Modern Approach, Prentice Hall.
3. B. K. P. Horn, Robot Vision, McGraw Hill.

Reference books

1. E. Trucco and A. Verri, Introductory Techniques for 3D Computer Vision, Publisher: Prentice Hall.
2. R. Jain et. Al, Machine Vision, McGraw Hill, 1995.
3. E. Trucco, and A. Verri, Introductory Techniques for 3-D Computer Vision, Prentice Hall, 1998.
4. V. Nawla, A Guided Tour of Computer Vision, Addison-Wesley, 1993.
5. Various journal and conference articles, research reports, and book excerpts as appropriate.

BCS-68 SOFTWARE REUSE

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. To provide a solid background knowledge about software Reuse.
2. To educate Metrics used in software reuse.
3. To provide Knowledge about various frameworks and COTS.

Topics Covered

UNIT-I

INTRODUCTION: Software Reuse and Software Engineering –State of Art and the Practice - Aspects of Software Reuse- Software Reuse Organizations – Support Services – Institutionalizing Reuse. 9

DOMAIN ENGINEERING: Building Reusable Assets – Domain Analysis: Basic Concepts – Domain Scoping – Domain vs Application Requirements – Domain Analysis Methods – Domain Analysis Tools- Programming Paradigms and Reusability.

UNIT-II

OBJECT ORIENTED DOMAIN ENGINEERING: A Pragmatic Introduction to Object Orientation: Introduction- the Tenets of Object Oriented Programming. Abstraction and Parameterization Techniques in Object Orientation: Abstraction Techniques in Object Oriented Modeling – Abstraction Techniques in Object Oriented Programming Languages, Meta-programming – Design Patterns. 9

UNIT-III

FRAMEWORKS AND APPLICATION ENGINEERING-Application Frameworks: Framework – Fulfilling the Framework Contract–Building Frameworks-SWING Framework. Architectural Frameworks: Architecture–Architecture and Reuse–CORBA – Application Engineering – Component Storage and Retrieval – Reusable Asset Integration. 9

UNIT-IV

MANAGERIAL ASPECTS OF SOFTWARE REUSE Software Reuse Metrics – Software Reuse Cost Estimation – Software Reuse Return on Investment – Component Based Software Engineering – Product-Line Engineering – COTS Based Development. 9

Textbooks

1. Hongji Yang and Xiaodong Liu, Software Reuse in the Emerging Cloud Computing Era, IGI Publishing Hershey, PA, USA, 2012.
2. Hafedh Mili, Ali Mili, Sherif Yacoub, Edward Addy, Reuse-Based Software Engineering: Techniques, Organizations, and Control", John Wiley &Sons, 2002.

Reference books

1. Carma McClure, Software Reuse: A Standards-Based Guide, IEEE, 2001.
2. Wayne C. Lim, Managing Software Reuse, Prentice Hall, 2004.
3. Ivar Jacobson, Martin Gres, Patrick Johnson, Software Reuse, Pearson Education, 2004.

BCS-69 SOFTWARE VERIFICATION & VALIDATION

Course Category : Program Elective (PE3 & PE4)
Pre-requisite Subject : NIL

Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand the concepts and theory related to software testing.
2. Understand different testing techniques used in designing test plans, developing test suites, and evaluating test suite coverage
3. Understand the relationship between black-box and white-box testing and know how to apply as appropriate.
4. Learn to use automated testing tools in order to measure code coverage.
5. Understand how software developers can integrate a testing framework into code development in order to incrementally develop and test code.

UNIT-I

An Introduction to Software Verification and Validation/Basic Concepts, Methods for Evaluating Software for Correctness and Reliability including Code, Inspections, Program Proofs, System Test Categories, Code inspections and their role in software verification. 9

UNIT-II

Review of Software Engineering Methods and Challenges, Role of Verification and Validation. Economics of Verification and Validation, Software Reviews and Inspections, Conducting Reviews and Inspection, Software Quality Metrics 9

UNIT-III

Review of Software Configuration Management, Software Testing Overview, Functional & Structural Testing, Integration and System Testing 9

UNIT-IV

Software validation metrics, Assessing and Improving the Validation Process, Improving the development Process 9

Textbooks

1. Stephen H Kan, Metric and Model in Software Quality Engineering, Pearson Education
2. William Perry, Effective methods for Software Testing, Wiley Publication
3. Dorotny Graham, Erik Van Veenendaal, Foundation of Software Testing By: CENGAGE learning,
4. Dr. K.V.K. Prasad, Software Testing Tools, Dreamtech Press
5. *Pankaj Jalote*, An Integrated Approach To *Software* Engineering, Springer Verlag, NY, 1991
6. Rajib Mall, Fundamentals of Software Testing, PHI Publication

Reference books

1. Steven R. Raktitin, Software Verification and Validation for Practitioners and Managers, ed. Artech House, 2nd Edition

BCS-70 SOFTWARE DESIGN & CONSTRUCTION

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand Architectural styles and Quality Attributes.
2. Understand common tools and terminology related to software design and construction.
3. Understand the role of the Software Architect with a development project.
4. Use methods for constructing and evaluating architectures.
5. Understand Advance Concepts in design and construction.

Topics Covered

UNIT-I

Software Architecture –Architecture Structures and Views – Importance of Software Architecture – Predicting System Quality-Influencing Organizational Structure – Improving Cost and Schedule estimates – Context of Software architecture. 9

UNIT-II

Understanding Quality Attributes – Availability – Interoperability – Modifiability - Performance and Security – Testability - Usability – Quality Attribute Modeling and Analysis. 9

UNIT-III

Architecture in Agile Projects – Architecture and Requirements – Designing and Documentation – Implementation and Testing – Architecture Reconstruction and Conformance. 9

UNIT-IV

Economic Analysis of Architecture – Architecture Competence – Architecture and Software Product Lines – Case Studies, Architecture in Cloud - Cloud Definition – Service Model – Economic Justification – Base Mechanism – Architecture for Edge – Edge Document System – SDLC – Metropolis Model. 9

Textbooks

1. Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, 3 rd edition Pearson, 2013.
2. Mary Shaw, David Garlan, Software Architecture: Perspectives on an Emerging Discipline, Prentice Hall, 1996.

Reference books

1. Taylor R. N, Medvidovic N, Dashofy E. M, Software Architecture: Foundations, Theory, and Practice, Wiley, 2009.
2. Booch G, Rumbaugh J, Jacobson I, The Unified Modeling Language User Guide, Addison-Wesley, 1999

BCS-71 SOFTWARE QUALITY MANAGEMENT

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Define quality assurance plans
2. Apply quality assurance tools & techniques
3. To learn about standards and certifications
4. To describe procedures and work instructions in software organizations

Topics Covered

UNIT-I

INTRODUCTION: Software Quality Challenge - Software Quality Factors - Components of the Software Quality Assurance System. Pre-Project Software Quality Components - Contract Review - Development and Quality Plans 9

UNIT-II

SQA COMPONENTS IN THE PROJECT LIFE CYCLE : Integrating Quality Activities in the Project Life Cycle – Reviews - Software Testing – Strategies - Software Testing – Implementation - Assuring the Quality of Software Maintenance - Assuring The Quality of External Participants' Parts - Case Tools and their effect on Software Quality. 9

UNIT-III

SOFTWARE QUALITY INFRASTRUCTURE COMPONENTS: Procedures and Work Instructions - Supporting Quality Devices - Staff Training, Instructing and Certification - 9

Preventive and Corrective Actions - Configuration Management - Documentation and Quality Records Controls.

UNIT-IV

SOFTWARE QUALITY MANAGEMENT COMPONENTS & STANDARDS: Project Progress Control - Components of Project Progress Control- Progress control of internal projects and external participants- Implementation of Project Progress Control, ISO 9001 Certification - Software Process Assessment. Organizing for Quality Assurance -Management and its Role in Quality Assurance - Software Quality Assurance Unit - SQA Trustees and Committees 9

Textbooks

1. Daniel Galin, Software Quality Assurance: From Theory to Implementation, Pearson Addison-Wesley, 2012.
2. Jeff Tian, Software Quality Engineering (SQE), Wiley-Interscience, 2005
3. Stephen H Kan, Metrics & Models in Software Quality Engineering, Pearson Education
4. Kshirasagar Naik & Priyadarshi Tripathi, Software Testing & Quality Assurance, Wiley India Edition
5. Stephen H. Kan, Metrics and models in software quality Engineering, Addison – Wesley, 1955.
6. Roger S. Pressman , Software Engineering-A Practitioner's Approach, McGraw Hill publication

Reference books

1. Mordechai Ben – Menachem and Garry S. Marliss, "Software Quality", Thomson Asia Pte Ltd, 2003.
2. Allen Gilles, "Software quality: Theory and management" - International Thomson - Computer press, 1997.

BCS-72 ASPECT ORIENTED PROGRAMMING

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. To master basics of aspect-oriented software development, this enables a higher degree of the separation of concerns through crosscutting concern modularization.
2. Provides an overview of aspect-oriented approaches to software development throughout all of its stages, as well as programming languages connected with these approaches. The course also covers the relationship of aspect-oriented software

development and software product lines.

3. Will gain experience with Aspect J, which is the most important aspect-oriented programming language of today.

Topics Covered

UNIT-I

Introduction to Aspect Oriented Programming, AOP Language Anatomy, AOP Language, Specification, AOP Language Implementation, AspectJ, Example Aspect, Obtaining and Installing AspectJ — 9

UNIT-II

Writing the Component First, Aspect Code, Identifying the Join Point, Determining the Point cut, Giving Advice, Adding an Aspect, Compiling and Executing the Example, Adding a New Concern, Exposing Context, Inter-type Declarations, Aspect Granularity, AspectJ Compiler Functionality — 9

UNIT-III

Introduction to AspectJ Joint Point: The Dynamic Join Point Model, AspectJ Join Points, Join Point Signatures, Patterns, Reflection, Example Join Points, AspectJ Pointcuts: Building Pointcuts, Using Designators, Combining Pointcuts — 9

UNIT-IV

Advice: Definition of Advice, Issues Common to All Types of Advice, Types of Advice: An Overview, Before Advice, After Advice, Around Advice, Advice Precedence, Inter type declarations, Aspects, Structure, Extensions, Aspect Instantiation and Associations, Use of AspectJ and its Tools, Error Handling and Common Problems — 9

Textbooks

1. Ivan Kiselev, Aspect-Oriented Programming with AspectJ, Sams, 2002.
2. Robert E. Filman, Tzilla Elrad, Siobhan Clarke, Mehmet Aksit, Aspect-Oriented Software Development, Addison-esley, 2004
3. Ivar Jacobson and Pan-Wei Ng. Aspect-Oriented Software Development with Use Cases. Addison-Wesley, 2004
4. Krzysztof Czarnecki and Ulrich Eisenecker. Generative Programming: Methods, Tools, and Applications. Addison-Wesley, 2000.

Reference books

1. Joseph D. Gradecki, Nicholas Lesiecki, Mastering AspectJ: Aspect Oriented programming in JAVA, Wiley, First Edition, 2003
2. O .Vladimir Safonov, Using Aspect-Oriented Programming for Trustworthy Software Development, John Wiley & Sons, 2008.
3. Siobhan Clarke and Elisa Baniassad. Aspect-Oriented Analysis and Design: The Theme Approach. Addison-Wesley, 2005.

BCS-73 NEURAL NETWORK & FUZZY SYSTEM

Course Category : Program Elective (PE3 & PE4)

Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Basics of ANN and its learning algorithms.
2. Fuzzy principles and relations.
3. Genetic algorithms and its applications.
4. Hybrid systems and usage of MATLAB toolbox

Topics Covered

UNIT-I

Neural Networks-1(Introduction &Architecture) Neuron, Nerve Structure and Synapse, Artificial Neuron and its Model, Activation Functions, Neural Network Architecture: Single Layer and Multilayer Feed Forward Networks, Recurrent Networks, Various Learning Techniques; Perception and Convergence Rule, Auto-Associative and Hetro-Associative Memory 9

UNIT-II

Neural Networks-II (Back Propagation Networks) Architecture: Perceptron Model, Solution, Single Layer Artificial Neural Network, Multilayer Perception Model; Back Propagation Learning Methods, Effect of Learning Rule Co-Efficient ;Back Propagation Algorithm, Factors Affecting Back-propagation Training, Applications. 9

UNIT-III

Fuzzy Logic-I (Introduction) Basic Concepts of Fuzzy Logic, Fuzzy Sets and Crisp Sets, Fuzzy Set Theory and Operations, Properties of Fuzzy Sets, Fuzzy and Crisp Relations, Fuzzy to Crisp Conversion, Membership Functions, Interference in Fuzzy Logic, Fuzzy If-Then Rules, Fuzzy Implications and Fuzzy Algorithms, Fuzzyfications & Defuzzificataions, Fuzzy Controller, Industrial Applications. 9

UNIT-IV

Genetic Algorithm(GA) Basic Concepts, Working Principle, Procedures of GA, Flow Chart of GA, Genetic Representations, (Encoding) Initialization and Selection, Genetic Operators, Mutation, Generational Cycle, Applications 9

Textbooks

1. S. Rajsekaran& G.A. Vijayalakshmi Pai, Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications, Prentice Hall of India.
2. N. P. Padhy, Artificial Intelligence and Intelligent Systems, Oxford University Press.
3. Siman Haykin, Neural Netw0rks, Prentice Hall of India

4. Timothy J. Ross, Fuzzy Logic with Engineering Applications, Wiley India.
5. S.N. Sivanandam & S.N. Deepa, Principles of Soft Computing, John Wiley & Sons, 01-Jun-2007

Reference books

1. Hertz J. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, California, 1991
2. Freeman J.A. & D.M. Skapura, Neural Networks: Algorithms, Applications and Programming Techniques, Addison Wesley, Reading, Mass, (1992).

BCS-74 FUNDAMENTALS OF CLOUD COMPUTING

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. understand the concept of Existing Hosting Platforms and computing paradigms currently being used in industry and academia
2. Identify the issues related to Cloud Computing. To analyse IASS/ PAAS and SAAS services along with Cloud models.
3. Understand the concepts of various Cloud Platforms with comparative analysis and the concepts of virtualization with the advantages in Cloud.

Topics Covered

UNIT-I

Introduction: Basics of Emerging Cloud Computing Paradigm, Cloud Computing History and Evolution, Cloud Enabling Technologies, Practical Applications of Cloud Computing for Various Industries, Economics and Benefits of Cloud Computing 9

Cloud Computing Architecture: Cloud Architecture Model, Types of Clouds: Public Private &Hybrid Clouds, Resource Management and Scheduling, QOS (Quality Of Service) and Resource Allocation, Clustering

UNIT-II

Classification of Cloud Implementations- Amazon Web Services - IaaS, Elastic Compute Cloud (EC2), Simple Storage Service (S3), Simple Queuing Services (SQS), VMware vCloud - IaaS, vCloud Express, Google AppEngine - PaaS, JAVA Runtime Environment 9

UNIT-III

Data Center : Classic Data Center, Virtualized Data Center (Compute, Storage, Networking and 9

Application) , Business Continuity in VDC

Virtualization: Virtualization, Advantages and disadvantages of Virtualization, Types of Virtualization: Resource Virtualization i.e. Server, Storage and Network virtualization, Migration of processes, VMware vCloud – IaaS

UNIT-IV

Cloud Security and Privacy: Infrastructure Security: Infrastructure Security: The Network Level, Infrastructure Security: The Host Level, Infrastructure Security: The Application Level, Data Security and Storage: Aspects of Data Security, Data Security Mitigation, Provider Data and Its Security. Privacy: Data Life Cycle, Key Privacy Concerns in the Cloud, Responsibility for Protecting Privacy, Changes to Privacy Risk Management and Compliance in Relation to Cloud Computing, Legal and Regulatory Implications 9

Textbooks

1. Dr. Kumar Saurabh, Cloud Computing, Wiley
2. Arshdeep Bahga, Vijay Madisetti, Cloud Computing: A Hands-on Approach, Universities Press
3. Tim Mather, Subra Kumaraswamy, Shahed Latif, Cloud Security and Privacy, O'Reilly Media

Reference books

1. Gerard Blokdijk, Ivanka Menken , The Complete Cornerstone Guide to Cloud Computing Best Practices, Second Edition, Emereo Pty Ltd, 2009
2. Anthony Velte, Toby Velte and Robert Elsenpeter, Cloud Computing: A practical Approach, Tata McGraw Hill
3. Raj Kumar Buyya, James Broberg, Andrezei M. Goscinski, Cloud Computing: Principles and Paradigms, , John Wiley and Sons2011
4. Michael Miller, Cloud Computing, Pearson Education India, 2008
5. Judith Hurwitz, Robin Bllor, Marcia Kaufmann, Fern Halper, Cloud Computing for Dummies, Wiley, 2009

BCS-75 ADVANCED MULTI-CORE SYSTEMS

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand the architectural techniques used in modern multi-core chips for mobile and server systems.
2. Understand the hardware support for security and parallel programming, and advanced memory systems.

3. Become acquainted with recent processor design techniques (superscalar cores, VLIW cores, multi-threaded cores, energy-efficient cores), cache coherence, memory consistency, vector processors, graphics processors, heterogeneous processors.
4. Exposing with complex trade-offs between performance-power-complexity, hardware-software interactions, and architecture-technology interactions.

Topics Covered

UNIT-I

Introduction to Multi-Core Architectures, Virtual Memory and Caches, Introduction to Parallel Programming, Cache Coherence and Memory Consistency Models, Hardware Support for Synchronization, Case Studies of Chip-Multiprocessor. 9

UNIT-II

Introduction to Program Optimization, Control-Flow Analysis, Data-Flow Analysis, Compilers for High-Performance Architectures, Data Dependence Analysis, Loop Optimizations. 9

UNIT-III

CPU Scheduling, OS Support for Synchronization, Multi-Processor Scheduling, Security Issues. 9

UNIT-IV

A Tutorial on OpenMP, A Tutorial on Intel Threading Tools. 9

Textbooks

1. J. L. Hennessy and D. A. Patterson. Computer Architecture: A Quantitative Approach. Morgan Kaufmann publishers.
2. D. E. Culler, J. P. Singh, with A. Gupta. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann publishers.

Reference books

1. Steven S. Muchnick. Advanced Compiler Design and Implementation. Morgan Kaufmann publishers.
2. Wolfe. Optimizing Supercompilers for Supercomputers. Addison-Wesley publishers.
3. Allen and Kennedy. Optimizing Compilers for Modern Architectures. Morgan Kaufmann publishers.
4. A. S. Tanenbaum. Distributed Operating Systems. Prentice Hall.
5. Coulouris, Dollimore, and Kindberg. Distributed Systems Concept and Design. Addison-Wesley publishers.
6. Silberschatz, Galvin, and Gagne. Operating Systems Principles. Addison-Wesley publishers.

BCS-76 CRYPTOGRAPHY AND INFORMATION SECURITY

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination

Course Outcomes : The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Encryption techniques and key generation techniques.
2. Authentication and security measures.
3. Intrusion and filtering analysis.

Topics Covered

UNIT-I

Introduction to Cryptography, Attacks, Services and Mechanism, Conventional Encryption Model, Classical Encryption Techniques- Substitution Ciphers and Transposition Ciphers, Cryptanalysis, Steganography, Stream and Block Ciphers, Modern Block Ciphers: Block Ciphers Principals, Data Encryption Standard (DES), Strength of DES, Differential and Linear Crypt Analysis of DES, Block Cipher Modes of Operations, Triple DES, IDEA Encryption and Decryption, Strength of IDEA, Confidentiality using Conventional Encryption, Traffic Confidentiality, Key Distribution, Random Number Generation. 9

UNIT-II

Introduction to Graph, Ring and Field, Prime and Relative Prime Numbers, Modular Arithmetic, Fermat's and Euler's Theorem, Euclid's Algorithm, Chinese Remainder Theorem. Principals of Public Key Crypto Systems, RSA Algorithm, Security of RSA, Key Management, Diffle-Hellman Key Exchange Algorithm, Elganel Encryption. 9

UNIT-III

Message Authentication and Hash Function: Authentication Requirements, Authentication Functions, Message Authentication Code, Hash Functions, Birthday Attacks, Security of Hash Functions and MACS, MD5 Message Digest Algorithm, Secure Hash Algorithm (SHA). Digital Signatures: Digital Signatures, Authentication Protocols, Digital Signature Standards (DSS), Authentication Applications: Kerberos, Electronic Mail Security-Pretty Good Privacy (PGP), S/MIME. 9

UNIT-IV

IP Security: Architecture, Authentication Header, Encapsulating Security Payloads, Combining Security Associations, Key Management. Web Security: Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction (SET), System Security: Intruders, Viruses and Related Threads, Firewall Design Principals, Trusted Systems. 9

Textbooks

1. William Stallings, Cryptography and Network Security: Principals and Practice, Pearson Publication.
2. Johannes A. Buchmann, Introduction to Cryptography, Springer-Verlag.
3. Bruce Schiener, Applied Cryptography, John Wiley and Sons, 1996
4. Behrouz A. Frouzan, Cryptography & Network Security, Tata McGraw Hill
5. Bruce Schiener, Applied Cryptography, John Wiley & Sons
6. Atul Kahate, "Cryptography and Network Security" Tata McGraw Hill

Reference books

1. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security, Private communication in public world, PHI Second Edition, 2002
2. Douglas R Simson, Cryptography – Theory and practice, CRC Press, First Edition, 1995

BCS-77 DIGITAL IMAGE PROCESSING

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. To understand Digital Image Processing fundamentals
2. To learn Image Transformation, Enhancement, Restoration and Compression Techniques.
3. To implement various techniques for Segmentation of Images
4. To learn the Image Reconstruction operations.
5. To implement Image Processing Techniques for suitable applications

Topics Covered

UNIT-I

Light, Brightness Adaptation and Discrimination, Pixels, Coordinate Conventions, Imaging Geometry, Perspective Projection, Spatial domain Filtering, Sampling and quantization. Intensity Transformations, Contrast Stretching, Histogram Equalization, Correlation and Convolution, 2-D Sampling, Discrete Cosine Transform, Frequency Domain Filtering. 9

UNIT-II

Transform, Fourier Transforms and Properties, FFT (Decimation in Frequency and Decimation in Time Techniques), Basic Framework, Interactive Restoration, Image Deformation and Geometric Transformations, Image Morphing, Restoration Techniques, Noise Characterization, Noise Restoration Filters, Adaptive Filters, Linear, Position Invariant Degradations, Estimation of Degradation Functions, Restoration from Projections. 9

UNIT-III

Types of Redundancies, Lossy and Lossless Compression, Entropy af An Information Source, Shannon's Theorem, Huffman Coding, Arithmetic Coding, Golomb Coding, Bit-Plane Encoding, Bit-Allocation, Zonal Coding, Threshold Coding, Lossless Predictive Coding, Lossy Predictive Coding, Motion Compensation Expansion of Functions, Multi Resolution Analysis, Scaling 9

Functions, Wavelet Series Expansion, Discrete Wavelet Transform (DWT), Continuous Wavelet Transform, Fast Wavelet Transform, 2-D Wavelet Transform, Digital Image Watermarking.

UNIT-IV

Basics of Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole Filling, Connected Components, Convex Hull, Thinning, Thickening, Skeletons, Pruning, Geodesic Dilation, Erosion, Reconstruction by Dilation and Erosion. Boundary Detection Based Techniques, Point, Line Detection, Edge Detection, Edge Linking, Local Processing, Regional Processing, Hough Transform, Thresholding, Iterative Thresholding, OTSU's Method, Moving Averages, Multivariable Thresholding, Region-Based Segmentation, Watershed Algorithm, Use of Motion in Segmentation 9

Textbooks

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, Second Edition, 2012.
2. Anil K. Jain, Fundamentals of Digital Image Processing, Pearson 2012.
3. Kenneth R. Castleman, Digital Image Processing, Pearson, 2011.
4. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins, Digital Image Processing using MATLAB, Pearson Education, Inc., 2010.
5. William K. Pratt, Digital Image Processing, John Wiley, New York, 2012.
6. D, E. Dudgeon and R M. Mersereau, Digital Signal Processing, Prentice Hall Professional Technical Reference, 2010.
7. Milan Sonka et al, Image Processing, Analysis and Machine Vision, Brookes/Cole,

Reference books

1. Jayaraman S., Esaki Rajan S., T.Veera Kumar, "Digital Image Processing", Tata McGraw Hill Pvt. Ltd., Second Reprint, 2010
2. Bhabatosh Chanda, Dwejesh Dutta Majumder, "Digital Image Processing and analysis", PHI Learning Pvt. Ltd., Second Edition, 2011
3. Malay K. Pakhira, "Digital Image Processing and Pattern Recognition", PHI Learning Pvt. Ltd., First Edition, 2011
4. Annadurai S., Shanmugalakshmi R., "Fundamentals of Digital Image Processing", Pearson Education, First Edition, 2007

BCS-78 HIGH PERFORMANCE COMPUTING

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand the role of HPC in science and engineering.
2. Become acquainted with the most commonly used HPC platforms and parallel programming models.
3. Become acquainted with the means by which to measure, analyze and assess the performance of HPC applications and their supporting hardware.
4. Develop mechanisms for evaluating the suitability of different HPC solutions to common problems found in Computational Science.
5. Perform the role of administration, scheduling, code portability and data management in an HPC environment, with particular reference to Grid Computing.
6. Understand potential benefits and pitfalls of Grid Computing.

Topics Covered

UNIT-I

Program Execution: Program, Compilation, Object Files, Function Call and Return, Address Space, Data And Its Representation. Computer Organization: Memory, Registers, Instruction Set Architecture, Instruction Processing. 9

UNIT-II

Pipelined Processors: Pipelining, Structural, Data and Control Hazards, Impact on Programming. Virtual Memory: Use of Memory by Programs, Address Translation, Paging, Cache Memory: Organization, Impact on Programming, Virtual Caches 9

UNIT-III

Operating Systems: Processes And System Calls, Process Management, Program Profiling, File Systems: Disk Management, Name Management, Protection 9

UNIT-IV

Parallel Architecture: Inter-Process Communication, Synchronization, Mutual Exclusion, Basics of Parallel Architecture, Parallel Programming with Message Passing using MPI 9

Textbooks

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar : Introduction to Parallel Computing, Second Edition Pearson Education, 2007
2. Michael J. Quinn, Parallel Programming in C with MPI and Open MP McGraw-Hill International Editions, Computer Science Series, 2004

Reference books

1. J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann.
2. Silberschatz, P. B. Galvin, G. Gagne, Operating System Concepts, John Wiley.
3. R. E. Bryant and D. R. O'Hallaron, Computer Systems: A Programmer's Perspective, Prentice Hall.

BCS-79 INTRODUCTION TO REAL TIME SYSTEM

Course Category	:	Program Elective (PE3 & PE4)
Pre-requisite Subject	:	NIL
Contact Hours/Week	:	Lecture : 3, Tutorial : 1 , Practical: 0
Number of Credits	:	4
Course Assessment Methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes and Three Minor tests and One Major Theory Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Real-time scheduling and schedulability analysis
2. Formal specification and verification of timing constraints and properties
3. Design methods for real-time systems
4. Development and implementation of new techniques to advance the state-of-the-art real-time systems research

Topics Covered

UNIT-I

Introduction- Issues in Real Time Computing, Structure of A Real Time System. Task Classes, Performance Measures for Real Time Systems, Estimating Program Run Times. Task Assignment and Scheduling - Classical Uniprocessor Scheduling Algorithms, Uniprocessor Scheduling of IRIS Tasks, Task Assignment, Mode Changes, and Fault Tolerant Scheduling. 9

UNIT-II

Programming Language and Tools – Desired Language Characteristics, Data Typing, Control Structures, Facilitating Hierarchical Decomposition, Packages, Run-Time (Exception) Error Handling, Overloading and Generics, Multitasking, Low Level Programming, Task Scheduling, Timing Specifications, Programming Environments, Run-Time Support. 9

UNIT-III

Real Time Databases - Basic Definition, Real Time vs General Purpose Databases, Main Memory Databases, Transaction Priorities, Transaction Aborts, Concurrency Control Issues, Disk Scheduling Algorithms, Two-Phase Approach to Improve Predictability, Maintaining Serialization Consistency, Databases for Hard Real Time Systems. 9

UNIT-IV

Real-Time Communication - Communications Media, Network Topologies Protocols, Fault Tolerant Routing. Fault Tolerance Techniques - Fault Types, Fault Detection. Fault Error Containment Redundancy, Data Diversity, Reversal Checks, Integrated Failure Handling. Reliability Evaluation Techniques - Obtaining Parameter Values, Reliability Models for Hardware Redundancy, Software Error Models. Clock Synchronization - Clock, A Non-Fault-Tolerant Synchronization Algorithm, Impact of Faults, Fault Tolerant Synchronization in 9

Hardware, Fault Tolerant Synchronization in Software

Textbooks

1. Alan C. Shaw, Real – Time Systems and software; John Wiley & Sons Inc
2. Rajib Mall, Real Time Systems: Theory and Practice by -Pearson Education, 2007
3. Jane W S Liu, Real Time Systems, Pearson

Reference books

1. Stuart Bennett, Real Time Computer Control-An Introduction", Second edition Prentice Hall PTR, 1994.
2. Peter D. Lawrence, Real time Micro Computer System Design – An Introduction, McGraw Hill, 1988
3. S.T. Allworth and R.N. Zobel, Introduction to real time software design", Macmillan, II Edition, 1987.
4. R.J.A Buhur, D.L. Bailey, An Introduction to Real-Time Systems", Prentice-Hall International, 1999.
5. Philip. A. Laplante Real Time System Design and Analysis" PHI, III Edition, April2004.
6. C.M. Krishna, Kang G. Shin, Real-Time Systems", McGraw-Hill International Editions, 1997.
7. Other materials required for the class will be made available during the course.

ANNEXURE-2

BCS-04A OBJECT ORIENTED MODELING & C++

Course Category	: Engineering Fundamental (EF)
Pre-requisite Subject	: NIL
Contact Hours/Week	: Lecture: 3, Tutorial: 1, Practical: 2
Number of Credits	: 5
Course Assessment Methods	: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	: The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course 1. Understand the Concept of Object Oriented Programming and Master OOP using C++ 2. Implementing the Real-Life Problems using Object Oriented Techniques. 3. Improvement in Problem Solving Skills.

UNIT-I **9**

Object Modeling: Objects and Classes, Links and Associations, Generalization and Aggregation, Metadata, Candidate Keys, Constraints, Dynamics Modeling: State and State Diagram, Functional Modeling: Data Flow Diagram

UNIT-II **9**

Object Oriented Programming: Features of Object Oriented Programming, C++ Fundamentals: data types, Operators and Expressions, Reference variables, Control flow, Arrays, Structures, Strings, Pointers and Functions, Overloading functions, Friend Function.

UNIT-III **9**

Defining Class, creating objects and accessing its member, Constructors and Destructors, Operator overloading and Type conversions, Inheritance and Polymorphism: Single inheritance, multi-level inheritance, multiple inheritance, hierarchical inheritance, runtime polymorphism, Virtual Functions and Abstract class.

UNIT-IV **9**

Templates and Exception Handling: Use of templates, function templates, class templates, handling exceptions. File handling: Stream in C++, Files modes, File pointer and manipulators, type of files, accepting command line arguments, Standard template library.

EXPERIMENTS

Write C++ Programs to illustrate the concept of the following:

1. Arrays
2. Structures
3. Pointers
4. Objects and Classes
5. Console I/O Operations
6. Scope Resolution and Memory Management Operators
7. Inheritance
8. Polymorphism
9. Virtual Functions
10. Friend Functions
11. Operator Overloading
12. Function Overloading
13. Constructors and Destructors
14. this Pointer
15. File I/O Operations

Analyse, Design and Develop Code for the Following System (one for a batch of three students) using C++

1. ATM (Automated Teller Machine) System
2. Contact Management System
3. Employee Record Management System
4. Stock Maintenance System
5. Course Registration System
6. Payroll System
7. Library Management System
8. Calendar

Textbooks

1. B. Trivedi Programming with ANSI C++, Oxford University Press, 2007.
2. Ira Pohl, Object Oriented Programming using C++, Pearson Education, Second Edition
3. B. Stroustrup, The C++ Programming Language, 3rdedition, Pearson Education, 2004
4. James Rumbaugh, et. al Object Oriented Modeling and Design-, PHI
5. Robert Lafore, Object Oriented Programming in Turbo C++, Galgotia Publication, 1994
6. E. Balaguruswamy, Object Oriented Programming with C++, TMH Publication
7. Grady Booch, James Rumbaugh and Ivar Jacobson The Unified Modeling Language User Guide, Pearson Education
8. Booch, Maksimchuk, Engle, Young, Conallen and Houstan, Object Oriented Analysis and Design with Applications, Pearson Education
9. S. B. Lippman, Josee Lajoie, Barbara E. Moo, C++ Primer,4th edition, Pearson Education, 2005

Reference Books

1. Coleman, D. et.al. Object-Oriented Development, The Fusion Method. Prentice Hall
2. Booch, G. Object-Oriented Design with Applications. Redwood City, Bengamin/Cummings
3. Sartaj Sahni, Data Structures, Algorithms and Applications in C++, McGraw Hill, Second Edition, 2005.

BCS-13A Internet & JAVA Programming

Course category	:	Department Core (DC)
Pre-requisites	:	NIL
Contact hours/week	:	Lecture: 3, Tutorial: 1, Practical: 2
Number of Credits	:	5
Course Assessment methods	:	Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination
Course Outcomes	:	The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. To identify different components of client server architecture on Internet computing.
2. Knowledge of how to develop and deploy applications and applets in JAVA.
3. Knowledge of how to develop and deploy GUI using Java Swing and AWT.
4. Design, develop and implement interactive web applications.
5. Be able to implement, compile, test and run JAVA programs comprising more than one class and to address a particular software problem.
6. To understand the basic concepts of Internet services and related technologies.
7. Develop programs using the JAVA Collection API as well as the JAVA standard class library.

UNIT-I

Internet: Introduction to Internet Services, Core Java: Introduction, Operator, Data type, Variables, Control Statements, Arrays, Methods & Classes, Constructors, String Handling, Inheritance, Package and Interface. 9

UNIT-II

Exception Handling, Multithread programming, I/O, Java Applet, Networking, Event handling, Introduction to AWT, AWT controls, Layout managers. 9

UNIT-III

Java Swing: Creating a Swing Applet, Labels, Text fields, Buttons, Tabbed Panes, JDBC: Connectivity Model, JDBC/ODBC Bridge, JAVA SQL package, connectivity to Remote Database, Remote method invocation (RMI). 9

UNIT-IV

Java Beans: Application Builder tools, The Bean Developer Kit(BDK), JAR files, Introspection, developing a simple bean, using Bound properties, The Java Beans API, Session Beans, Entity Beans, Introduction to Java Servlet: Servlet Basics, Servlet API basic, Life cycle of a Servlet, Running Servlet. 9

EXPERIMENTS

1. Basic programs of simple statements, conditional statements, iterative statement, and arrays.
2. Programs having object-oriented concepts like Inheritance and Interface.
3. Programs for Exception Handling and Event Handling.
4. Programs of Threads and Multithreading.
5. Programs related to Applets and Swings.
6. Program including JAVA Beans and Servlets.

Textbooks

1. Naughton, Schildt, "The Complete Reference JAVA2", TMH.
2. Balagurusamy E, "Programming in JAVA", TMH

Reference Books

1. Margaret Levine Young, "The Complete Reference Internet", TMH.
2. Dustin R. Callway, "Inside Servlets", Addison Wesley.
3. Mark Wutica, "Java Enterprise Edition", QUE.
4. Steven Holzner, "Java2 Black book", Dreamtech.

BCS-80 PROGRAMMING IN C

Course Category: Engineering Fundamental (EF)
Pre-requisite Subject: NIL
Contact Hours/Week: Lecture: 3, Tutorial: 1, Practical: 2
Number of Credits: 5
Course Assessment Methods: Continuous assessment through tutorials, attendance, home assignments, quizzes, practical work, record, viva voce and Three Minor tests and One Major Theory & Practical Examination.
Course Outcomes: The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Basic Terminology used in Computer Programming.
2. Programs Development in C Language by Writing, Compiling and Debugging.
3. Design of Programs involving Simple Statements, Conditional Statements, Iterative Statements, Array, Strings, Functions, Recursion, Structure and Union.
4. Difference between Call by Value and Call by Reference.
5. Dynamic Memory Allocations and Use of Pointers.
6. Basic Operations on a File.
7. Basics of Dynamic Memory.

UNIT-I **9**

Basics of Programming: Approaches to Problem Solving, Concept of Algorithm and Flow Charts, Types of Computer Languages-Machine Language, Assembly Language and High-Level Language, Concept of Assembler, Compiler, Loader and Linker. Data types, Operators, Expressions, Operator Precedence and Associativity.

Fundamentals of C Programming: Structure of C Program, Writing and Executing the First C Program, Components of C Language, Standard I/O, Formatted I/O.

Conditional Program Execution: Applying if and switch Statements, Nesting if and else.

Program Loops and Iterations: Use of while, do while and for Loops, Multiple Loop Variables, Use of break and continue Statements, goto Statement.

UNIT-II **9**

Arrays: One Dimensional, Multidimensional Array and Their Applications, Declaration and Manipulation of Arrays.

Strings: String Variable, String Handling Functions, Array of Strings.

Functions: Designing Structured Programs, Functions in C, User Defined and Standard Functions, Formal vs. Actual Arguments, Function Category, Function Prototype, Parameter Passing, Recursive Functions.

Storage Classes: Auto, Extern, Register and Static.

UNIT-III **9**

Pointers: Pointer Variable and its Importance, Pointer Arithmetic Pointers and Arrays, Pointer and Character Strings, Pointers and Functions, Array of Pointers, Pointers to Pointers.

Structure: Declaration and Initialization of Structures, Structure as Function Parameters, Structure Pointers.

Union: Declaration and Initialization of Unions, Union as Function Parameters, Union Pointers.

UNIT-IV **9**

Dynamic Memory Allocation: malloc, calloc, realloc, free function.

File Management: Defining and Opening a File, Closing a File, Input/ Output Operations in Files, Random Access to Files, Error Handling.

The Pre-processor Directives, Macros, Command Line Arguments, Introduction to Graphics Programming.

EXPERIMENTS

1. Write programs to print statements in sequential order using simple printf, scanf input/output functions.
2. Write programs to implement *if-else* condition (simple as well as nested) on suitable problems.
3. Write program to implement *switch-case* conditional logic on suitable examples.
4. Write programs to implement *for, while* and *do-while* loop control statements on suitable problems.
5. Write programs to implement 1D & 2D array concepts on suitable problems such as sorting of elements, searching of element, matrix addition, subtraction, multiplication etc.
6. Write programs to implement string related concepts such as sorting of a string, finding its length, reversing, concatenation, comparing two strings etc.
7. Write programs to implement concept of user defined functions (call by value, call by reference, recursive calling etc.) on suitable examples.
8. Write programs to implement concepts of pointer.
9. Write programs to implement the concept of structure and union.
10. Write programs to implement dynamic memory allocation functions (calloc, malloc, free, realloc)
11. Write programs to implement file handling concepts such as reading from a file, writing to a file using file related functions (fclose, fopen, sscanf, sprint, fread, fwrite, getc, putc, getw, putw etc.)

Textbooks

1. Jeri R. Hanly and Elliot B. Koffman, Problem Solving and Program Design in C, 7th Edition, Pearson.
2. Schildt, Herbert, Complete Reference with C, Tata McGraw Hill.
3. Kerninghan and Ritchie, The C programming Language, 2nd Edition, Prentice Hall.
4. Richard Bird, Introduction to Functional Programming using Haskell, 2nd Edition, Prentice-Hall International, 1998.

Reference Books

1. Greg Michaelson, An Introduction to Functional Programming Through Lambda Calculus, Dover Edition, Addition Wesley Publication.
2. Samuel P. Harbison, and Guy L. Steele Jr., C-A Reference Manual, Fifth Edition, Prentice Hall, 2002.