Babu Banarasi Das University, Lucknow

Department of Computer Science & Engineering

School of Engineering

Bachelor of Technology

Evaluation Scheme

SEMESTER III									
Course Category	**Course Code**	**Code Title**	**Contact Hours**			**Evaluation Scheme**			**Credits**
			L	**T**	**P**	**CIA**	**ESE**	**Course Total**	
C	BHS2301/ BHS2302	Industrial Psychology / Industrial Sociology	2	0	0	40	60	**100**	**2**
C	BAS2301	Complex Analysis and Integral Transforms	3	1	0	40	60	**100**	**4**
C	BCS2301	Operating System	3	1	0	40	60	**100**	**4**
C	BCS2302	Object Oriented Analysis and Design (Using UML)	2	1	0	40	60	**100**	**3**
C	BCS2303	Digital Logic Design	3	1	0	40	60	**100**	**4**
C	BCS2305/ BCS2405	Programming in ‘C’	3	1	0	40	60	**100**	**4**
C	BCS2351	Operating Systems Lab	0	0	2	40	60	**100**	**1**
C	BCS2352	Object Oriented Programming Lab	0	0	2	40	60	**100**	**1**
C	BCS2353	Digital Logic Design Lab	0	0	2	40	60	**100**	**1**
C	BCS2355/ BCS2455	‘C’ Programming Lab	0	0	2	40	60	**100**	**1**
	GP2301	General Proficiency	-	-	-	100	-	**100**	**1**
Total			**16**	**5**	**8**	**500**	**600**	**1100**	**26**

SEMESTER IV									
Course Category	**Course Code**	**Code Title**	**Contact Hours**			**Evaluation Scheme**			**Credits**
			L	**T**	**P**	**CIA**	**ESE**	**Course Total**	
C	BHS2402/ BHS2401	Industrial Sociology /Industrial Psychology	2	0	0	40	60	**100**	**2**
C	BAS2401	Statistical and Numerical Techniques	2	1	0	40	60	**100**	**3**
C	BCS2401	Database Management Systems	3	1	0	40	60	**100**	**4**
C	BCS2402	Discrete Mathematics	3	1	0	40	60	**100**	**4**
C	BCS2403	Data Structure Using 'C'	3	1	0	40	60	**100**	**4**
C	BCS2404	Computer Organization & Architecture	3	1	0	40	60	**100**	**4**
C	BCS2451	Database Management System Lab	0	0	2	40	60	**100**	**1**
C	BCS2452	Numerical Technique Lab	0	0	2	40	60	**100**	**1**
C	BCS2453	Data Structure Lab	0	0	2	40	60	**100**	**1**
C	BCS2454	Computer Organization & Architecture Lab	0	0	2	40	60	**100**	**1**
	GP2401	General Proficiency	-	-	-	100	-	**100**	**1**
Total			**16**	**5**	**8**	**500**	**600**	**1100**	**26**

SEMESTER V									
Course Category	Course Code	Code Title	Contact Hours			Evaluation Scheme			Credits
			L	T	P	CIA	ESE	Course Total	
C	BHS2501	Engineering & Managerial Economics	3	0	0	40	60	**100**	**3**
C	BCS2501	Software Engineering	3	1	0	40	60	**100**	**4**
C	BCS2502	Microprocessor and Interfacing	3	1	0	40	60	**100**	**4**
C	BCS2503	Computer Networks	3	1	0	40	60	**100**	**4**
C	BCS2504	Automata theory	3	1	0	40	60	**100**	**4**
C	BCS2505	Computer Graphics	2	1	0	40	60	**100**	**3**
C	BCS2552	Microprocessor and Interfacing Lab	0	0	2	40	60	**100**	**1**
C	BCS2553	Computer Networks Lab	0	0	2	40	60	**100**	**1**
C	BCS2555	Computer Graphics Lab	0	0	2	40	60	**100**	**1**
	GP2501	General Proficiency	-	-	-	100	-	**100**	**1**
Total			**17**	**5**	**6**	**460**	**540**	**1000**	**26**

SEMESTER VI									
Course Category	Course Code	Code Title	Contact Hours			Evaluation Scheme			Credits
			L	T	P	CIA	ESE	Course Total	
C	BHS2601	Industrial Management	3	0	0	40	60	**100**	**3**
C	BCS2601	Core and Advance JAVA	3	1	0	40	60	**100**	**4**
C	BCS2602	Design & Analysis of Algorithms	3	1	0	40	60	**100**	**4**
C	BCS2603	Advanced Computer Architecture	2	1	0	40	60	**100**	**3**
C	BCS2604	Compiler Design	3	1	0	40	60	**100**	**4**
GE		**Generic Elective I**	3	1	0	40	60	**100**	**4**
C	BCS2651	JAVA Lab	0	0	2	40	60	**100**	**1**
C	BCS2652	Design & Analysis of Algorithms Lab	0	0	2	40	60	**100**	**1**
C	BCS2658	Seminar	0	0	2	100	0	**100**	**1**
	GP2601	General Proficiency	-	-	-	100	-	**100**	**1**
Total			**18**	**4**	**6**	**520**	**480**	**1000**	**26**

Note: The students need to undergo a 4 to 6 weeks of industrial training that will be evaluated in the VII Semester.

SEMESTER VII									
Course Category	**Course Code**	**Code Title**	**Contact Hours**			**Evaluation Scheme**			**Credits**
			L	**T**	**P**	**CIA**	**ESE**	**Course Total**	
C	BCS2701	Web Technology	3	1	0	40	60	**100**	**4**
C	BCS2702	Network Security and Cryptography	3	1	0	40	60	**100**	**4**
GE		**Generic Elective II**	3	1	0	40	60	**100**	**4**
GE		**Generic Elective III**	3	1	0	40	60	**100**	**4**
OE		**Open Elective I***	-	-	-	40	60	**100**	**4**
C	BCS2751	Web Tech Lab	0	0	2	40	60	**100**	**1**
C	BCS2752	Network Security Lab	0	0	2	40	60	**100**	**1**
C	BCS2759	Project I[#]	0	0	2	100	0	**100**	**1**
C	BCS2758	Industrial Training Evaluation	0	0	2	100	0	**100**	**1**
	GP2701	General Proficiency	-	-	-	100	-	**100**	**1**
Total			**12**	**4**	**8**	**580**	**420**	**1000**	**25**

*Students will opt any one of the open elective from the list of open electives provided by the university

[#]Students need to submit an abstract for the project, select a guide and will complete at least 20% of the project work.

SEMESTER VIII									
Course Category	**Course Code**	**Code Title**	**Contact Hours**			**Evaluation Scheme**			**Credits**
			L	**T**	**P**	**CIA**	**ESE**	**Course Total**	
C	BCS2801	Distributed System	**3**	**0**	**0**	40	60	**100**	**3**
GE		**Generic Elective IV**	**3**	**1**	**0**	40	60	**100**	**4**
GE		**Generic Elective V**	**3**	**1**	**0**	40	60	**100**	**4**
OE		**Open Elective II****	**-**	**-**	**-**	40	60	**100**	**4**
C	BCS2851	Distributed System Lab	**0**	**0**	**2**	40	60	**100**	**1**
C	BCS2859	Project II[##]	**0**	**0**	**16**	160	240	**400**	**8**
	GP2801	General Proficiency	**-**	**-**	**-**	100	-	**100**	**1**
Total			**9**	**2**	**18**	**460**	**540**	**1000**	**25**

**The opted subject should be different from the one selected in VII Semester

[##]This is in continuation with the project work started in Semester VII. In this semester the students will complete the project

Legends:

L Number of Lecture Hours per week
T Number of Tutorial Hours per week
P Number of Practical Hours per week
CIA Continuous Internal Assessment
ESE End Semester Examination

Category of Courses:

F Foundation Course
C Core Course
GE Generic Elective
OE Open Elective

List of Open Electives Offered by Department of Computer Science and Engineering

S.N.	Course Code	Open Elective
1	OCS2001	Human Computer Interaction
2	OCS2002	E-Commerce

Course Code	Generic Elective I
BCS2011	Multimedia Systems
BCS2012	Soft Computing
BCS2013	Evolutionary Algorithms
BCS2014	Distributed Database Systems

Course Code	Generic Elective II
BCS2021	System Programming & System Administration
BCS2022	System Modeling & Simulation
BCS2023	Artificial Intelligence
BCS2024	Embedded System Design

Course Code	Generic Elective III
BCS2031	Cyber law and Security
BCS2032	Real Time Operating System
BCS2033	Robotics
BCS2034	Computer Vision

Course Code	Generic Elective IV
BCS2041	Data Mining and Ware Housing
BCS2042	Artificial Neural Network (ANN) and Fuzzy Logic
BCS2043	Mobile Computing
BCS2044	Natural Language Processing

Course Code	Generic Elective V
BCS2051	Data Compression
BCS2052	Digital Image Processing
BCS2053	Bioinformatics
BCS2054	Pattern Recognition

Credit Summary Chart										
Course Category	**Semester**								**Total Credits**	**%age**
	I	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**		
F	16	10	0	0	0	0	0	0	26	12.44
C	10	17	25	25	25	21	12	12	147	70.33
GE	0	0	0	0	0	4	8	8	20	9.56
OE	0	0	0	0	0	0	4	4	8	3.82
GP	1	1	1	1	1	1	1	1	8	3.82
Total	**27**	**28**	**26**	**26**	**26**	**26**	**25**	**25**	**209**	**100**

Discipline Wise Credit Summary Chart										
Course Category	**Semester**								**Total Credits**	**%age**
	I	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**		
Basic Sciences	10	12	4	3					**29**	13.87
Humanities &Socials Sciences	0	5	2	2	3	3			**15**	7.17
Engg. Sciences	16	10							**26**	12.44
Professional Subject Core			19	20	22	17	10	4	**92**	44.01
Professional Subject - General Elective						4	8	8	**20**	9.56
Professional Subject - Open Elective							4	4	**8**	3.82
GP + Project Work, Seminar and / or Internship in Industry or elsewhere.	1	1	1	1	1	2	3	9	**19**	9.09
Total	**27**	**28**	**26**	**26**	**26**	**26**	**25**	**25**	**209**	**100**

BCS2301 Operating system

Course Objective:

1. Study the basic concepts and functions of operating systems.
2. Understand the structure and functions of OS.
3. Learn about Processes, Threads and Scheduling algorithms.
4. Understand the principles of concurrency and Deadlocks.
5. Learn various memory management schemes.
6. Study I/O management and File systems.

Learning Outcome:

At the end of the course, the student should be able to:

1. Design various Scheduling algorithms.
2. Apply the principles of concurrency.
3. Design deadlock, prevention and avoidance algorithms.
4. Compare and contrast various memory management schemes.
5. Design and Implement a prototype file systems.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction to Operating System and Process Concept** Operating system and functions, Classification of Operating systems, Operating System Structure, Operating System Services, System call and System program, Process concept, Process state, Process control block, Context switching, Operation on process, Threads and their management, Benefits of multithreading, Types of threads, Threading issues, CPU-scheduling, Scheduling criteria, Scheduling Algorithms, Concurrent Processes, Inter Process Communication models and Schemes	30 Hours	1
II	**Process Synchronization and Deadlock** Process synchronization, Producer/Consumer Problem, Critical Section Problem, Peterson's solution, Synchronization of hardware, Semaphore, Classical-problem of synchronization, Deadlock, Deadlock characterization, Deadlock Prevention, Deadlock Avoidance, Resource allocation graph algorithm, Banker's algorithm, Deadlock detection, Recovery from deadlock	30 Hours	1
III	**Memory Management** Memory Management, Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing	30 Hours	1

IV	**I/O Management And File System** File System Structure**,** File System Implementation**,** Directory Implementation and Allocation Methods**,** Free space Management**,** Kernel I/O Subsystems, Disk Structure, Disk Scheduling, Disk Management**,** Swap-Space Management	30 Hours	1

Text/Reference Books:

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

BCS2302 Object Oriented Analysis and Design (Using UML)

Course Objective:

1. Study the basic concepts and functions of UML.
2. Learn about Modelling, Architecture and SDLC.
3. Learn various Diagrams like Class and Object diagram.
4. Understand the Modelling Techniques.
5. Learn various Architectural modelling likes activity and component diagrams.

Learning Outcome:

At the end of the course, the student should be able to:

1. Draw the various Classes and Objects.
2. Apply the Modelling Techniques.
3. Use various Relationships.
4. Compare various interaction Diagrams.
5. Design and implement activity, component and deployment diagram.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction to UML** Introduction to UML Model, Importance of Model, Principles of Modelling, and Object oriented Modelling, Conceptual Model of UML, Architecture, and Software Development Life Cycle.	30 Hours	1
II	**Basic and Advanced Structural Modeling** Classes and Advanced Classes, Relationship and Advanced Relationship – Dependency, Generalisation and Realization, Association and Aggregation- Association Class, Interfaces, Packages. Class and Object Diagrams-Terms and Concepts and Modelling Techniques. **Basic Behavioral Modeling-I** Interactions- Terms and Concepts, Interaction diagrams and its Type - Sequential diagram and	30 Hours	1

	Collaboration diagram, Modeling Techniques.		
III	**Basic Behavioral Modeling-II** Use cases- Terms and Concepts, Use case diagrams- Use case and Collaboration and Modeling Techniques, Activity diagrams-Forking and Joining, Swimlanes, Object flow. **Architectural Modeling** Events and signals, Processes and Threads, State Machines, State Chart diagrams, Component and Component diagrams- Terms and Concepts and Modeling Techniques. Deployment and Deployment diagrams- Terms and Concepts and Modeling Techniques.	30 Hours	1

Text/Reference Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson: "The Unified Modeling Language User Guide", Pearson Education.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: "UML 2 Toolkit", WILEY -Dreamtech India Pvt. Ltd.
3. Meilir Page-Jones: "Fundamentals of Object Oriented Design in UML", Pearson Education.
4. Pascal Roques, "Modeling Software Systems Using UML2", WILEY-Dreamtech India Pvt. Ltd.
5. Atul Kahate, "Object Oriented Analysis & Design", McGraw-Hill.
6. Mark Priestley, "Practical Object-Oriented Design with UML", McGraw Hill

BCS2303 Digital Logic Design

Course Objective:

1. Introduce the concept of digital and binary systems
2. Be able to design and analyze combinational logic circuits.
3. Be able to design and analyze sequential logic circuits.
4. Reinforce theory and techniques taught in the classroom through experiments and projects in the laboratory.

Learning Outcome:

At the end of the course, the student should be able to:

1. Define different number systems, binary addition and subtraction, 2's complement representation and operations with this representation.
2. Understand the different switching algebra theorems and apply them for logic functions.
3. Define the Karnaugh map for a few variables and perform an algorithmic reduction of logic functions.
4. Define the following combinational circuits: buses, encoders/decoders, (de)multiplexers, exclusive-ORs, comparators, arithmetic-logic; and to be able to build simple applications.
5. Understand the bistable element and the different latches and flip-flops.
6. Derive the state-machine analysis or synthesis and to perform simple projects with a few flip-flops.

7. Understand sequential circuits, like counters and shift registers, and to perform simple projects with them.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Binary Numbers and Gate level Minimization** Digital system and binary numbers: Signed binary numbers, Binary codes, Cyclic codes, Error detecting and correcting codes, Hamming codes, Floating point representation. Gate-level minimization: The map method up to five variable, Don't care conditions, Karnaugh maps, POS simplification, NAND and NOR implementation, Quine Mc¬Clusky method (Tabular method).	30 Hours	1
II	**Combinational Logic** Combinational circuits: analysis procedure, design procedure; Binary adder-subtractor; decimal adder; binary multiplier; magnitude comparator; decoders; encoders; multiplexers.	30 Hours	1
III	**Synchronous Sequential Logic** Sequential circuits: Binary cell, Latches, Flip-flops-RS, JK, Master-Slave JK, D F/F,T flip flops; Synchronous Sequential Circuit Design: Fundamentals of Synchronous sequential circuits; Classification of synchronous machines; Analysis of Synchronous Sequential circuits; Design of Synchronous and Asynchronous Counters-Shift registers, Ring counters; Analysis and design of Finite State Machines; Timing issues in synchronous circuits.	30 Hours	1
IV	**Asynchronous Sequential Logic** Fundamentals of Asynchronous Sequential circuits; Analysis and design of Asynchronous Sequential circuits; Pulse mode and Fundamental-mode Circuits; Cycles; Races; Hazards in asynchronous circuits; Programmable Logic Devices: PLAs, PALs, CPLD; FPGA Architecture; Finite state machines: Mealy, Moore design; Introduction to VHDL; Examples of System design applications like Washing machine, Candy Vending machine, traffic lights.	30 Hours	1

Text/Reference Books:

1. Raj Kamal, "Digital Systems Principle and Design", Pearson Education.
2. Balbaniam, Carison, "Digital Logic Design Principles", Wiley Publications.
3. Morris Mano, "Digital Design", Pearson Education.
4. R.P. Jain, "Modern Digital Electronics", McGraw Hill.
5. D.P. Leach, A .P. Malvino, "Digital Principles and Applications", TMH.
6. Tocci, Digital systems, "Principles and applications", Pearson.

7. Sudhakar Yalamachili, John M. Yarbrough, “Introductory VHDL”, Pearson Education.

BCS2305/BCS2405 Programming in ‘C’

Course Objective:

1. Is to acquaint the students with the basics of computers system.
2. Its components, data representation inside computer and to get them familiar with various important features of procedure oriented programming language i.e. C.
3. To learn C programs and to apply your skills in writing real application programs using C.
4. Explains all the concepts of C programming language clearly with simple programs.
5. To learn the basics of all other programming languages such as C++, Java,

Learning Outcome:

At the end of the course, the student should be able to:

1. Understand the basic terminology used in computer programming.
2. Write, compile and debug programs in C language.
3. Use different data types in a computer program.
4. Design programs involving decision structures, loops and functions.
5. Explain the difference between call by value and call by reference
6. Understand the dynamics of memory by the use of pointers.
7. Use different data structures and create/update basic data files.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction** Programming Environment, Concept of algorithm, Strategy for designing Algorithms, Top-down development, Stepwise refinement, Flowchart, Programming Languages, Assembler, Compiler, Interpreter, Systematic Development of Programs, Program Writing and execution, Introduction to the design and implementation of correct efficient and maintainable programs, Structured Programming Concept, Number System and Conversion Methods, Introduction to C language, Identifiers, Keywords, Constants and Variables in C, Fundamental Data types in C, Integer types, short, long. Unsigned Character types, single and double precision floating point.	30 Hours	1
II	**Storage Classes, Operators and Control Statements** Storage Classes in C: Automatic, register, static, extern, Operators and Expressions in C: Arithmetic, Relational, Logical, Assignment, Bitwise,	30 Hours	1

	Conditional, Increment and Decrement, Special Operators such as comma, sizeof etc. Type Conversion in C, Operator Precedence and Associativity, Mixed mode operations, Standard Input/output functions: printf(), scanf(), getch(), getchar(), getche() etc. Conditional and Control Statements: if statement, if-else statement, nested if-else statement, else if ladder, switch statements, restrictions on switch values, Use of break and default statement with switch. Looping or Iteration: Uses of while, for and do-while loops, nesting of loops, use of break and continue statements.		
III	**Arrays, Structures and Pointers** Array, notation and representation, using one dimensional, two dimensional and multi-dimensional arrays, Arrays of unknown and varying size, Searching and sorting in arrays. Strings: String declaration and initialization, String manipulation. Structures: Purpose and use of structures, declaring and assigning of structures, accessing structure elements, Array of structures, Arrays within structures. Union: Utility of unions, Union of structures. Pointers: Understanding Pointers, Declaration and initialization of pointer variables, Accessing the address of the variable, Pointer arithmetic, Pointers and arrays. Dynamic Memory Allocation, Stack, Linked list	30 Hours	1
IV	**Functions, Preprocessors and File Handling** Function Declaration, function Definition, function call, Passing values between functions, Global and local variables and their scope, Call by value and call by reference ,Recursion, Pointers to functions, Declaration of a pointer to a function, Initialization of function pointers, Calling a function using a function pointer, Passing a function to another function, How to return a function pointer. Standard C library functions: Math functions, String handling functions, The C preprocessor: preprocessor directives, defining and calling macros, conditional compilation, passing values to the compiler. File Handling in C: Types of files, Defining, opening and closing of a file, Input/output operations on files, Multiple file handling in C.	30 Hours	1

Text/Reference Books:

1. Yashavant P. Kanetkar, ”Let Us ‘C’”, BPB Publications
2. Jeri R. Hanly, Elliot B. Koffman, ”Problem Solving and Program Design in C”, Pearson Addison-Wesley.
3. Behrouz A., ”Computer Science- A Structured Programming Approach Using C”.

4. E Balaguruswamy, ”Computer Concepts and Programming in C”, Tata McGraw Hill Publications

BCS2351 Operating Systems Lab

Note: Minimum 8 experiments to be performed by students

List of Experiments:

Implement the following on LINUX or other Unix like platform. Use C for high level language implementation

1. Write programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, close, stat, open dir, read dir
2. Write programs using the I/O system calls of UNIX operating system (open, read, write, etc)
3. Write C programs to simulate UNIX commands like ls, grep, etc.
4. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.
5. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.
6. Developing Application using Inter Process communication (using shared memory, pipes or message queues).
7. Implement the Producer – Consumer problem using semaphores (using UNIX system calls).
8. Implement memory management scheme: Free space is maintained as a linked list of nodes with each node having the starting byte address and the ending byte address of a free block. Each memory request consists of the process-id and the amount of storage space required in bytes. Allocated memory space is again maintained as a linked list of nodes with each node having the process-id, starting byte address and the ending byte address of the allocated space.
9. Implement any file allocation technique (Linked, Indexed or Contiguous).

BCS2352 Object Oriented Programming Lab

Tools used: Violet UML Editor/Agro-UML / MS Visio / Rational Rose

Language used: UML 2.0

1. Develop a UML Class diagram For Online railway reservation system.
2. Develop a UML Class diagram for Library Management system.
3. Develop a Use-Case Model for Student Admission System in a University, make necessary assumptions.
4. Develop Use-Case diagram for a library, where a member can perform two operations issue book and return it. A book is issued to a member only after verifying his credentials
5. Develop Use-Case diagram for a travel agent.
   To book a flight ticket the travel agent need to know the details about journey date and user address. The user can pay either by cash or by card. User can

also cancel a booked ticket later Travel agent also allows to book a hotel along with flight ticket, so while cancelling a flight ticket the agent also cancel hotel booking. Appropriate refund as per policy is made in case of cancellation.

6. Develop a UML class diagram & sequence diagram for a web browser consisting of different sub-components, which can be primarily categorized into browser rendering engine and browser control. The web browser control, too, consists of several sub-components including navigation, window control, event handlers, page display etc.
7. Develop a UML Sequence diagram for an information system for online book store.
8. Develop a UML Class diagram & Sequence diagram for car manufacturing process.
9. Develop State-Transition diagram for online air reservation system
10. Develop Deployment diagram for different scenarios of Railway Reservation System.

Text/Reference Books:

1. Use Case Driven Object Modelling with UML: Theory and Practice, Apress, 2007
2. UML 2 Certification Guide: Fundamental & Intermediate Exams (The MK/OMG Press), Morgan Kaufmann, 2006
3. UML Distilled: A Brief Guide to the Standard Object Modelling Language, Addison Wesley

BCS2353 Digital Logic Design Lab

Note: Minimum 10 experiments to be performed by students

List of Experiments:

1. Simplification, Realization of Boolean Expressions using Logic gates/Universal gates.
2. Realization of Half/Full adder and Half/Full Subtractors using logic gates.
3. Realization of parallel adder/Subtractors using 7483 chip.
4. BCD to Excess-3 code conversion and vice versa.
5. Realization of Binary to Gray code conversion and vice versa.
6. MUX/DEMUX–use of 74153, 74139 for arithmetic circuits and code converter.
7. Realization of One/Two bit comparator and study of 7485 magnitude comparator.
8. Use of a Decoder chip to drive LED display and Priority encoder.
9. Truth table verification of Flip-Flops:
    a. JK Master slave (ii) T type (iii) D type.
10. Realization of 3 bit counters as a sequential circuit and MOD–N counter design (7476, 7490, 74192, 74193).
11. Perform Shift left; Shift right, SIPO, SISO, PISO, PIPO operations using 74S95.
12. Wiring and testing of Ring counter/Johnson counter.
13. Wiring and testing of Sequence generator.

BCS2355/ BCS2455 'C' Programming Lab

Note: Minimum 10 experiments to be performed by students.

List of Experiments: Using Turbo C / Visual Studio 6.0 C++ environment

Part I:

1. Creating and editing simple C program, debugging, compilation, execution.
2. 'C' programming on variables and expression assignment, simple arithmetic Loops, If-else, Case statements, break, continue, goto.
3. Implementing different operations on Single & Multidimensional arrays.
4. Implementing different String handling inbuilt and user defined functions.
5. Functions, recursion, file handling in 'C'.
6. Pointers, address operator, declaring pointers and operations on pointers.
7. Address of an array, structures, pointer to structure, dynamic memory allocation.

BCS2401 Database Management Systems

Course Objective:

1. To Study the fundamental concepts of Database Management.
2. To develop skill of Database Design, Database Languages and Database-System implementation with respect to Relational Database Management System.
3. To develop the concepts of Transaction Processing System , Concurrency control, and Recovery procedures in database.

Learning Outcome:

At the end of the course, the student should be able to:

1. Understand the basic concepts of the database and data models.
2. Design a database using ER diagrams and map ER into Relations and normalize the relations.
3. Develop a simple database applications using normalization.
4. Acquire the knowledge about different special purpose databases and to critique how they differ from traditional database systems.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Database System Concepts, Database Users, and Architecture** Introduction to Database System with example, Characteristics of the Database Approach, Users of Database System, Advantages and disadvantages of Using a DBMS, Implications of the Database Approach, Data Models, Schemas, and Instances, DBMS Architecture and Data Independence, Database Languages and Interfaces, The Components of Database System, Classification of Database Management Systems	30 Hours	1

II	**Data Modelling & Relational Database Management System** Data Modelling Using the Entity-Relationship Model, concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Entity Types, Entity Sets, and Attributes, Relationships, Relationship Types, Roles, and Structural Constraints, Strong vs Weak Entity Types, ER Diagrams, Naming Conventions, and Design Issues, Enhanced Entity-Relationship Modelling, Subclasses, Super classes, and Inheritance, Specialization and Generalization, Constraints and Characteristics of Specialization and Generalization, Modelling of UNION Types Using Categories, The Relational Data Model, Relational Constraints, and the Relational Algebra, Relational Model Concepts, Relational Constraints and Relational Database Schemas, Update Operations and Dealing with Constraint Violations, Basic Relational Algebra Operations, Additional Relational Operations, Examples of Queries in Relational Algebra	30 Hours	1
III	**SQL and Database Design Theory and Methodology** Structured Query Language- The Relational Database Standard, Data Definition, Constraints, and Schema Changes in SQL, Types of SQL Commands, SQL Operators and their Procedure, Insert, Delete, and Update Statements in SQL, Queries and Sub Queries, Aggregate Functions, Joins, Unions, Intersection, Minus, Views (Virtual Tables) in SQL, Cursors, Triggers and PL/SQL, Functional Dependencies and Normalization for Relational Databases, Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form	30 Hours	1
IV	**Transaction Processing, Concurrency Control and Database Recovery** Transaction Processing Concepts, Introduction to Transaction Processing, Transaction states and State Diagram, Transaction and System Concepts, Desirable Properties of Transactions, Schedules and Recoverability, Serializability of Schedules, Concurrency Control Techniques, Locking Techniques for Concurrency Control, Concurrency Control Based on Timestamp Ordering, Multiversion Concurrency Control Techniques, Validation	30Hours	1

	(Optimistic) Concurrency Control Techniques, Granularity of Data Items and Multiple Granularity Locking, Database Recovery Techniques, Recovery Concepts, Recovery Techniques Based on Deferred Update, Recovery Techniques Based on Immediate Update, Shadow Paging, The ARIES Recovery Algorithm, Database Backup and Recovery from Catastrophic Failures		

Text/Reference Books:

1. Korth, Silbertz, Sudarshan, "Data Base Concepts", McGraw-Hill.
2. Ivan Bayross, "SQL, PL/SQL: The Programming Language Of Oracle"
3. Date C. J., "An Introduction to Data Base System", Addision Wesley.
4. Elmasri, Navathe, "Fundamentals Of Data Base Systems", Addision Wesley.
5. Bipin C. Desai, "An introduction to Data Base Systems", Galgotia Publication.
6. Ramakrishnan, Gehrke, 'Data Base Management System", McGraw-Hill.
7. Connolly &Begg, "Database Systems: A Practical Approach to Design, Implementation and Management", Pearson Education.
8. R. S. Despandey, "SQL/PL SQL for Oracle'.

BCS2402 Discrete Mathematics

Course Objective:

1. To introduce a number of Discrete Mathematical Structures (DMS) found to be serving as tools even today in the development of theoretical computer science.
2. Course focuses on of how Discrete Structures actually helped computer engineers to solve problems occurred in the development of programming languages.
3. Also, course highlights the importance of discrete structures towards simulation of a problem in computer science and engineering.
4. Introduction of a number of case studies involving problems of Computer Technology.

Learning Outcome:

At the end of the course, the student should be able to**:**

1. Have a complete knowledge on various discrete structures available in literature.
2. Have realization of some satisfaction of having learnt that discrete structures are indeed useful in computer science and engineering and thereby concluding that no mistake has been done in studying this course.
3. Gain some confidence on how to deal with problems which may arrive in computer science and engineering in near future.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Set Theory, Relations, Functions & Natural Numbers** Set Theory: Introduction, Combination of sets,	30 Hours	1

	Multisets, Ordered pairs, Proofs of some general identities on sets. Relations: Definition, Operations on relations, Properties of relations, Composite Relations, Equality of relations, Recursive definition of relation, Order of relations. Functions: Definition, Classification of functions, Operations on functions, Recursively defined functions, Growth of Functions. Natural Numbers: Introduction, Mathematical Induction, Variants of Induction, Induction with Nonzero Base cases, Proof Methods, Proof by counter – example, Proof by contradiction.		
II	**Groups, Rings, Fields & Lattice** Algebraic Structures: Definition, Groups, Subgroups and order, Cyclic Groups, Cosets, Lagrange's theorem, Normal Subgroups, Permutation and Symmetric groups, Group Homomorphisms. Definition and elementary properties of Rings and Fields, Integers Modulo n; Partial order sets: Definition, Partial order sets, Combination of partial order sets, Hasse diagram. Lattices: Definition, Properties of lattices ,Bounded, Complemented, Modular, Complete lattice;	30 Hours	1
III	**Boolean Algebra & Proposition Logic** Boolean Algebra: Introduction, Axioms and Theorems of Boolean algebra, Algebraic manipulation of Boolean expressions, Simplification of Boolean Functions, Karnaugh maps, Logic gates, Digital circuits and Boolean algebra. Propositional Logic: Proposition, well-formed formula, Truth tables, Tautology, Satisfiability; Contradiction; Algebra of proposition; Theory of Inference; Predicate Logic: First order predicate-well-formed formula of predicate, quantifiers, Inference theory of predicate logic.	30 Hours	1
IV	**Trees, Graph, Recurrence Relation & Combinatorics** Trees : Definition: Binary tree, Binary tree traversal, Binary search tree; Graphs: Definition and terminology: Representation of graphs, Multigraphs, Bipartite graphs, Planar graphs, Isomorphism and Homeomorphism of graphs,Euler and Hamiltonian paths, Graph coloring. Recurrence Relation & Generating function: Recursive definition of functions, Recursive algorithms, Method of solving recurrences. Combinatorics: Introduction; Counting Techniques: Pigeonhole Principle, Pólya's Counting Theory.	30 Hours	1

Text/Reference Books:

1. Koshy, "Discrete Structures", Elsevier Publication.
2. Kenneth H. Rosen, "Discrete Mathematics and Its Applications", McGraw-Hill.
3. B. Kolman, R.C. Busby, and S.C. Ross, "Discrete Mathematical Structures", Prentice Hall.
4. R.P. Grimaldi, "Discrete and Combinatorial Mathematics", Addison Wesley.
5. Jean Paul Trembley, R Manohar, "Discrete Mathematical Structures with Application to Computer Science," McGraw-Hill.

BCS2403 Data Structure using 'C'

Course Objective:

1. Demonstrate familiarity with major algorithms and data structures.
2. Analyse performance of algorithms.
3. Choose the appropriate data structure and algorithm design method for a specified application.
4. Determine which algorithm or data structure to use in different scenarios.
5. Be familiar with writing recursive methods.
6. Demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, trees and graphs.
7. Use various data structures effectively in application programs.
8. Demonstrate understanding of various sorting algorithms, including bubble sort, insertion sort, selection sort, heap sort and quick sort.
9. Understand and apply fundamental algorithmic problems including Tree traversals, Graph traversals, and shortest paths.
10. Demonstrate understanding of various searching algorithms.

Learning Outcome:

At the end of the course, the student should be able to:

1. Demonstrate knowledge of underlying data structures needed for solving problems and programming.
2. Implement/ utilize various data structures using a programming language such as C.
3. Analyze algorithms in connection with data structures.
4. Demonstrate knowledge of various searching and sorting techniques.
5. Apply dynamic memory allocation in creation of linked lists.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction** Introduction: Basic Terminology, Data types and its classification, Algorithm complexity notations like big Oh, Time-Space trade- off. Abstract Data Types (ADT). Array: Array Definition, Representation and Analysis of Arrays, Single and Multidimensional Arrays, Address calculation, Array as Parameters, Sparse Matrices, Recursion- definition and processes,	30 Hours	1

	simulating recursion, Backtracking, Recursive algorithms, Tail recursion, Removal of recursion, Tower of Hanoi		
II	**Stack and Linked List** Stack, Array Implementation of stack, Linked Representation of Stack, Application of stack: Conversion of Infix to Prefix and Postfix Expressions and Expression evaluation, Queue, Array and linked implementation of queues, Circular queues, D-queues and Priority Queues. Linked list, Implementation of Singly Linked List, Two-way Header List, Doubly linked list, Linked List in Array. Generalized linked list, Application: Garbage collection and compaction, Polynomial Arithmetic	30 Hours	1
III	**Tree, Searching, Sorting and Hashing** Trees: Basic terminology, Binary Trees,, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Binary Search Tree(BST), AVL Trees, B-trees. Application: Algebraic Expression, Huffman coding Algorithm. Internal and External sorting, Insertion Sort, Bubble Sort, selection sort, Quick Sort, Merge Sort, Heap Sort, Radix sort, Searching &Hashing: Sequential search, binary search, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation. Symbol Table, Static tree table, Dynamic Tree table.	30 Hours	1
IV	**Graphs** Graphs: Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi-list, Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm.	30 Hours	1

Text/Reference Books:

1. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C and C++", PHI.
2. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publication.
3. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with applications", McGraw Hill.
4. R. Kruseetal, "Data Structures and Program Designin C", Pearson Education
5. Lipschutz, "Data Structures ", Schaum's Outline Series, TMH.
6. GAV Pai, "Data Structures and Algorithms", TMH.

BCS2404 Computer Organization & Architecture

Course Objective: To learn

1. How Computer Systems work & its basic principles.
2. How to analyse the system performance.
3. Concepts behind advanced pipelining techniques.
4. The current state of art in memory system design.
5. How I/O devices are being accessed and its principles.

Learning Outcome:

At the end of the course, the student should be able to:

1. Apply the knowledge of performance metrics find the performance of systems.
2. Create an assembly language program for a microprocessor system.
3. Design a hardware component for an embedded system.
4. Deal with different types of computers.
5. Identify high performance architecture design.
6. Identify the problems in components of computer.
7. Develop independent learning skills and be able to learn more about different computer architectures and hardware.
8. Learn & use the new technologies in computers.
9. Use the knowledge of micro programming in the field of speech processing.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Computer Evolution & Arithmetic** A Brief History of computers: Designing for Performance, Von Neumann Architecture, Hardware architecture, Computer Components, Interconnection Structures, Bus Interconnection, Register Transfer Language, Bus and Memory Transfers, Bus Architecture, Bus Arbitration Techniques, Arithmetic Logic, Shift Micro operation, Arithmetic Logic Shift Unit, Design of Fast address, Arithmetic Algorithms (addition, subtraction, Booth Multiplication, Division)	30 Hours	1
II	**Control Unit** Control Design: Hardwired & Micro Programmed , Register Transfers, Performing of arithmetic or logical operations, Fetching a word from memory, Storing a word in memory, Execution of a complete instruction, Multiple-Bus organization, Hardwired Control, Micro programmed control, Microinstruction, Micro program sequencing, Wide-Branch addressing, Microinstruction with Next-address field, Prefetching, Microinstruction, Pipeline control: Instruction pipelines, Pipeline performance	30 Hours	1
III	**Processor Organization and Input-Output Organization** Processor Design: General register organization, Stack organization, Addressing mode, Instruction	30 Hours	1

	format, Data transfer & manipulations, Program Control, Reduced Instruction Set Computer, Input-Output Organization: Peripheral devices, I/O interface I/O ports. Interrupts: Interrupt hardware, Types of interrupts and exceptions, Modes of Data Transfer: Programmed I/O, Interrupt initiated I/O, Direct Memory Access: I/O channels and processors, Serial Communication: Synchronous & asynchronous communication, Standard communication interfaces		
IV	Memory Organization: Basic concept and hierarchy, semiconductor RAM memories,2D & 2 1/2D memory organization, ROM memories, Cache memories, Concept and design issues 9 performance, Address mapping and replacement, Auxiliary memories: Magnetic disk, Magnetic tape and optical disks, Virtual memory, Concept implementation	30 Hours	**1**

Text/Reference Books:

1. Patterson, "Computer Organisation and Design", Elsevier Publication
2. William Stalling, "Computer Organization", PHI.
3. Vravice, Hamacher & Zaky, "Computer Organization", TMH.
4. Moris Mano, " Computer System Architecture", PHI
5. John P Hays, " Computer Organization", McGraw Hill
6. Tannenbaum," Structured Computer Organization", PHI
7. P Pal Chaudhry, " Computer Organization & Design", PHI

BCS2451 Database Management System Lab

Note: Minimum 8 experiments to be performed by students

List of Experiments:

1. Write the queries for Data Definition and Data Manipulation Language.
2. Write SQL queries using logical operations (=, <,>, etc)
3. Write SQL queries using SQL operators.
4. Write SQL query using character, number, date and group functions.
5. Write SQL queries for relational algebra.
6. Write SQL queries for extracting data from more than one table.
7. Write SQL queries for sub queries, nested queries.
8. Write programme by the use of PL/SQL.
9. Concepts for ROLL BACK, COMMIT & CHECK POINTS.
10. Create VIEWS, CURSORS and TRGGERS & write ASSERTIONS.
11. Create FORMS and REPORTS.

BCS2452 Numerical Technique Lab

Write Programs in 'C' Language:

1. To deduce error evolved in polynomial equation.
2. To Find out the root of the Algebraic and Transcendental equations using Bisection, Regula-falsi, Newton Raphson and Iterative Methods. Also give the rate of convergence of roots in tabular form for each of these methods.
3. To implement Newton's forward and Backward Interpolation formula.
4. To implement Gauss Forward and Backward, Bessel's, Sterling's and Evertt's Interpolation formula
5. To implement Newton's Divided Difference and Langranges Interpolation formula.
6. To implement Numerical Differentiations.
7. To implement Numerical Integration using Trapezoidal, Simpson 1/3 and Simpson 3/8 rule.
8. To implement Least Square Method for curve fitting.
9. To draw frequency chart like histogram, frequency curve and pie-chart etc.
10. To estimate regression equation from sampled data and evaluate values of standard deviation, t-statistics, regression coefficient, value of R2 for at least two independent variables.

BCS2453 Data Structure Lab

List of Experiments:

Write Programs in C or C++ for following.

1. Array implementation of Stack, Queue, Circular Queue, List.
2. Implementation of Stack, Queue, Circular Queue, List using Dynamic memory Allocation.
3. Implementation of Tree Structures, Binary Tree, Tree Traversal, Binary Search Tree, Insertion and Deletion in BST.
4. Implementation of Searching and Sorting Algorithms.
5. Graph Implementation, BFS, DFS, Minimum cost spanning tree, shortest path algorithm.

BCS2454 Computer Organization & Architecture Lab

Note: Minimum 10 experiments to be performed by students

List of Experiments: Using Xilinx ISE Design tool and VHDL as the Hardware Description Language

1. Design and verify the behaviour of all 2-input basic gates (AND, OR, NOT).
2. Design and verify the behaviour of 2-input NAND, NOR, XOR gates.
3. Design and verify the behaviour of 2-input AND, OR, NOT, XOR gates with the help of NAND and NOR gates.
4. Design a half adder circuit with gate level description.
5. Design a full adder with gate level description.
6. Design a 4 bit BP
7. A (Binary Parallel Adder) with gate level description.

8. Design a 2 to 4 decoder using gate level description.
9. Design and simulate a 2 × 1 MUX.
10. Design and simulate an 8 × 1 MUX.
11. Design and simulate a 3 to 8 decoder.
12. Design and simulate a 1 bit comparator.
13. Design and simulate a 4 bit comparator.
14. Design and simulate a half subtractor.
15. Design and simulate a full subtractor.
16. Design and simulate a full adder.

BCS2501 Software Engineering

Course Objective:

1. Study the basic concepts and functions of Software Engineering.
2. Learn about Software Requirement Specification.
3. Learn various Designing and Testing techniques.
4. Understand the Techniques of Debugging.
5. Learn various Software project planning activities
6. Study Software Quality & Maintenance.

Learning Outcome:

At the end of the course, the student should be able to:

1. Selection and implementation of different software development process models.
2. Extracting and analyzing software requirements specifications for different projects.
3. Developing some basic level of software architecture/design.
4. Applying standard coding practices.
5. Defining the basic concepts and importance of Software project management concepts like cost estimation, scheduling and reviewing the progress.
6. Identification and implementation of the software metrics.
7. Applying different testing and debugging techniques and analyzing their effectiveness.
8. Analyzing software risks and risk management strategies.
9. Defining the concepts of software quality and reliability on the basis of International quality standards.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	Introduction to Software Engineering, Software Engineering Approaches, Defining the Problem, Developing a solution strategy, Planning and development process, Other planning activities, Software Life Cycle Models, Visual Modelling, Software Development Methods, Software Cost Estimation, Staffing Level Estimation, Estimating Software Maintenance Cost	30 Hours	1
II	Software Requirement Specification, Formal Requirement Speciation, Formal Requirement Verification, Axiomatic Specification, Algebraic Specification	30 Hours	1
III	Software Design Paradigms, Function Oriented Software Design, Object Oriented Design, UML, Design Patterns, User Design Interface Pattern, Coding, Software Testing, Unit Testing, Integration and System Testing, Debugging Techniques	30 Hours	1
IV	Software Quality & Maintenance, Software quality standards, SEI CMM and ISO-9001, Software reliability and fault-tolerance, Software Project	30 Hours	1

	Management,Software project planning,Software monitoring and control,Software maintenance,Computer-aided software engineering (CASE),Software reuse,Component-based software development,Extreme programming		

Text/Reference Books:

1. Rajib Mall, Fundamentals of Software Engineering, Prentice Hall India.
2. PankajJalote, An integrated approach to Software Engineering, Springer/Narosa.
3. Roger S. Pressman, Software Engineering: A practitioner's approach, McGraw Hill.
4. Ian Sommerville, Software Engineering, Addison-Wesley.

BCS2502 Microprocessor & Interfacing

Course Objective:

1. The course deals with applications, organization, architecture and design of microprocessors systems
2. To implement interfacing from a microprocessors based system to peripheral devices
3. learn how the hardware and software components of a microprocessor-based system work together to implement system-level features;
4. learn both hardware and software aspects of integrating digital devices (such as memory and I/O interfaces) into microprocessor-based systems;
5. learn the operating principles of, and gain hands-on experience with, common microprocessor peripherals such as UARTs, timers, and analog-to-digital and digital-to-analog converters;
6. get practical experience in applied digital logic design and assembly-language programming; and
7. Be exposed to the tools and techniques used by practicing engineers to design, implement, and debug microprocessor-based systems (during the Lab).

Learning Outcome:

At the end of the course, the student should be able to:

1. Design and conduct experiments related to microprocessor based system design and to analyze their outcomes.
2. Design, debug and test a small scale microprocessor system.
3. Identify, formulate, and solve engineering problems in microprocessor based system design.
4. Identify, formulate, and solve engineering problems in microprocessor based system design.
5. Use design tools for microprocessor system design, test and evaluation.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction: Memory & Microprocessor** Introduction: Memory Unit-Primary Memory: RAM, SRAM, DRAM, ROM, PROM, EPROM and EEPROM, Secondary Memory: Magnetic Memory,	30 Hours	1

	Tap, disc; cache memory**;** Real and virtual memory; Semiconductors Technology for memory; Addressing capacity of CPU; processing speed of computer; Shift registers; Evolution of Microprocessor and its types; Microprocessor architecture and operation of its components; Addressing modes ;Interrupts; Data transfer schemes; Instruction and data flow; Timer and timing diagram; Interfacing devices**;** Architectural advancement of microprocessors; Typical microprocessor development schemes.		
II	**8085 Microprocessor** 8 bit Microprocessor: Internal architecture; PIN diagram; interrupt and machine cycle; Instruction sets: Addressing modes; Instruction Classification; machine control and assembler directives; Technical features of: The Pentium, PentiumPro Micro Processor, Pentium II, Pentium III, Pentium – IV Microprocessor.	30 Hours	1
III	**8086 Microprocessor & Assembly Language Programming** 16-bit Microprocessor: Architecture of 8086 microprocessor, Register organization, Bus interface unit, Execution unit, Memory addressing, Operating modes; Instruction sets: Instruction format, Types of instructions; Introduction to 8086 family: Procedure and macros, connection, Timing and Troubleshooting, Interrupts; 80286, 80836 and 80486 microprocessor system concept. Programming: Assembly language programming based on Intel 8085/8086: Instructions- data transfer, arithmetic, logic, branch operations, looping, counting, indexing, programming techniques, counters and time delays, stacks, subroutines, Conditional calls and returns instructions; Introduction to Debugging program; Modular programming; Structured programming; Top-down; Bottom-up design; MACRO microprogramming.	30 Hours	1
IV	**Peripheral Interfacing** Introduction to Peripheral Devices 8237 4.2 DMA Controller;8255 programmable peripheral interface; 8253/8254 programmable timer/counter**;** 8259 programmable interrupt controller; 8251 USART and RS232C.	30 Hours	1

Text/Reference Books:

1. Gaonkar, Ramesh S: "Microprocessor Architecture, Programming and Applications with 8085", Penram International Publishing.
2. Ray A K , Bhurchandi K M : "Advanced Microprocessors and Peripherals", TMH
3. Hall D V: "Microprocessor Interfacing", TMH

4. Liu and Gibson G A: "Microcomputer System: The 8086/8088 family'', PHI
5. Aditya P Mathur: "Introduction to Microprocessor", TMH
6. Brey, Barry B: "INTEL Microprocessors", PHI
7. Renu Singh & B. P. Singh, "Microprocessor, Interfacing and Applications", 8. M Rafiqzzaman, "Microprocessors, Theory and Applications", Prentice Hall, 1992

BCS2503 Computer Networks

Course Objectives:

1. Build an understanding of the fundamental concepts of computer networking.
2. To understand the organization of computer networks, factors influencing computer network development and the reasons for having variety of different types of networks.
3. To apply knowledge of different techniques of error detection and correction to detect and solve error bit during data transmission.
4. To demonstrate proper placement of different layers of ISO model and illuminate its function.
5. To understand internals of main protocols such as FTP, SMTP, TCP, UDP, IP
6. To analyze simple protocols and can independently study literature concerning computer networks.

Learning Outcomes:

At the end of the course, the student should be able to:

1. Independently understand basic computer network technology.
2. Identify the different types of network topologies and protocols.
3. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.
4. Identify the different types of network devices and their functions within a network
5. Understand and building the skills of routing mechanisms.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction** Introduction: Network objectives and applications; network structure and architecture; OSI reference model; network services; network standardization; examples of networks; Physical layer: Fundamentals of data communication; transmission media; analog transmission; digital transmission; switching; ISDN; terminal handling; Broadcast channels and medium access: LAN protocols; CSMA with collision detection; collision free protocols; IEEE standard 802 for LANs; comparison of LANs; fibre optic networks	30 Hours	1

	and FDDI.		
II	**Data link layer and network layer** Data link layer: Design issues; error detection and corrections; elementary data link protocols; sliding window protocols. Examples; Network layer: Design issues; routing algorithms; congestion control; internetworking. Examples.	30 Hours	1
III	**Transport, Session and Presentation layer** Transport layer: Design Issues; connection management; example of a simple transport protocol. Session layer: Design issues; remote procedure call; examples, Presentation layer: Design issues; data compression and encryption; network security and privacy. Examples;	30 Hours	1
IV	**Application Layer** Application layer: Design issues; File transfer and file access; electronic mail; virtual terminals; other applications, Case study based on available network software.	30 Hours	1

Text/Reference Books:

1. Andrew S. Tanenbaum "Computer Networks" Prentice Hall of India.
2. William Stallings "Local Networks" Maxwell Macmillan International Edition.
3. B.A. Forouzan "Data Communication And Networking". Tata McGraw Hill.

BCS2504 Automata Theory

Course Objective:

1. Study the basic concepts and functions of Grammar
2. Study the basic concepts of finite automata
3. Learn about Regular expression and conversion methods
4. Understand the Push down automata.
5. Design various Turing machine

Learning Outcome:

At the end of the course, the student should be able to:

1. Design various language by DFA and NFA
2. Uses of Regular expression.
3. Design the CFG and CFL and PDA
4. Compare CNF and GNF.
5. Design and Implement Turing machine.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction to Automata** Introduction, Alphabets, Strings and Languages**,** Automata and Grammars**,** Deterministic finite	30 Hours	1

	Automata (DFA)-Formal Definition- State transition graph, Transition table, Language of DFA. Nondeterministic finite Automata (NFA)-NFA with epsilon transition, Language of NFA, Equi valence of NFA and DFA, Minimization of Finite Automata, Myhill-Nerode Theorem.		
II	**Regular expression** 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 and Decision properties of Regular Languages, FA with output -Moore and Mealy machine, Equivalence of Moore and Mealy Machine.	30Hours	1
III	**Context free grammar and Context Free Languages** Definition, Examples, Derivation , Derivation trees ,Ambiguity in Grammar, -Inherent ambiguity , Ambiguous to Unambiguous CFG, Useless symbols ,Simplification of CFGs, Normal forms for CFGs: CNF and GNF, Closure and Decision properties of CFL, Pumping lemma for CFLs. **Push Down Automata (PDA)**-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.	30 Hours	1
IV	**Turing machines (TM):** Basic model, definition and representation, Language acceptance by TM, Variants of Turing Machine, 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 and Introduction to recursive function theory .	30 Hours	1

Text/Reference Books:

1. 1.Hopcroft, Ullman, "Introduction to Automata Theory, Languages and Computation",
   Pearson Education .
2. K.L.P. Mishra and N. Chandrasekaran, "Theory of Computer Science : Automata, Languages and Computation", PHI Learning Private Limited, Delhi India.
3. Peter Linz, "An Introduction to Formal Language and Automata", Narosa Publishing house.

4. Y. N. Singh “Mathematical Foundation of Computer Science”, New Age International.
5. Papadimitrou, C. and Lewis, C.L., “Elements of the Theory of Computation”, PHI Learning Private Limited, Delhi India.
6. K. Krithivasan and R. Rama; Introduction to Formal Languages, Automata Theory and Computation; Pearson Education.
7. Harry R. Lewis and Christos H. Papadimitriou, Elements of the theory of Computation, Second Edition, Prentice-Hall of India Pvt. Ltd.
8. Micheal Sipser, “Introduction of the Theory and Computation”, Thomson Learning

BCS2505 Computer Graphics

Course Objective:

1. Giving students an introduction about computer graphics
2. Give a brief about Line and circle drawing algorithm
3. Learn 2D Clipping algorithms
4. Learn 3-D Object representation
5. Giving Introduction to Curves and Surfaces

Learning Outcome:

At the end of the course, the student will be able to:

1. Design various Line and Circle drawing algorithms.
2. Design various 2D clipping algorithms.
3. Understand 3-D Object Representation
4. Design Back Face Detection algorithm

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**INRODUCTION TO COMPUTER GRAPHICS, GRAPHIC DISPLAYS, LINE AND CIRCLE DRAWING ALGORITHM** Introduction to computer graphics, Advantages and application of computer graphics, Classification of computer graphics Graphic displays: Random scan displays, Raster scan displays, Line drawing algorithms: DDA algorithm, Bresenham’s algorithm, Parallel line algorithm, Circle drawing algorithm: DDA circle drawing algorithm, Bresenham’s circle drawing algorithm , Midpoint circle drawing algorithm	30 Hours	1
II	**TRANSFORMATIONS, WINDOWING AND CLIPPING AND 3-D TRANSFORMATION** Basic transformation: Translation**,** Rotation**,** Scaling**,** Matrix representations**,** Homogenous coordinates**,** Composite transformations Reflections and shearing transformations, Viewing Pipeline	30Hours	1

	Viewing transformations, 2D Clipping algorithms, Cohen Sutherland line clipping algorithm, Liang Barsky algorithm, Line clipping against non-rectangular clip windows, Polygon Clipping, Sutherland Hodgeman polygon clipping, Weiler and Atherton polygon clipping, Curve clipping and Text clipping, 3-D geometric primitives, 3-D Object representation, 3-D Transformation, 3-D viewing and projections, 3-D Clipping		
III	**Unit 3:CURVES AND SURFACES, HIDDEN LINES AND SURFACES AND BASIC ILLUMINATION MODEL** Introduction to Curves and Surfaces**,** Quadric surfaces**,** Spheres, Ellipsoid, Blobby objects**,** Introductory concepts of Spline**,** Bspline and Bezier curves and surfaces**,** Introduction to Hidden Lines and Surfaces**,** Back Face Detection algorithm**,** Depth buffer method**,** A- buffer method**,** Scan line method**,** basic illumination models Ambient light, Diffuse illumination, Specular reflection and Phong model, Combined approach, Warn model, Intensity Attenuation, Colour consideration, Transparency and Shadows	30 Hours	1

Text/Reference Books:

1. Donald Hearn and M Pauline Baker, "Computer Graphics C Version", Pearson Education
2. Amrendra N Sinha and Arun D Udai," Computer Graphics", TMH
3. Donald Hearn and M Pauline Baker, "Computer Graphics with OpenGL", Pearson education
4. Steven Harrington, "Computer Graphics: A Programming Approach", TMH
5. Rogers, "Procedural Elements of Computer Graphics", McGraw Hill

BCS2552 Microprocessor and Interfacing Lab

Note: Minimum 8 experiments to be performed by students

List of Experiments:

1. To study 8085 microprocessor System.
2. To study 8086 microprocessor System.
3. To develop and run a program to find out largest and smallest number.
4. To develop and run a program for converting temperature from F to C degree.
5. To develop and run a program to compute square root of a given number.
6. To develop and run a program for computing ascending/descending order of a number.
7. To perform interfacing of RAM chip to 8085/8086.
8. To perform interfacing of keyboard controller.
9. To perform interfacing of DMA controller.
10. To perform interfacing of UART/USART.

BCS2553 Computer Networks Lab

Tool used: The Network Simulator - ns-2

I. Access Devices:

1. Network Adaptor Drivers: Installation and Comparison
2. Tracing Linux Adapter Drivers (*)
3. Sniffing and Analyzing Protocol Messages (*)
4. Simulating Medium Access Control Protocols
5. Physically Wiring and Connecting LANs
6. Configuring WLAN and Mobile IP

II. Core Devices:

1. Unix Router: Setup, Testing, and Tracing ()
2. Probing Internet: Path and Delay Measurements
3. Configuring Cisco Routers
4. Subnetting: Configuration and Tracing in Linux
5. Benchmarking L2/L3 Switches with Smartbits

III. Edge/Server Proxy Devices:

1. IP Tools: Tracing finger and fingerd
2. Building Intranet with Linux
3. Building Firewall with Linux
4. Building Servers with Linux
5. Building Anti-Virus and Anti-Spam Gateway with Linux
6. Building Intrusion Detection and Prevention Gateway with Linux

BCS2555 Computer Graphics Lab

Tools/software used: Turbo C/ OpenGL

1. Implementation of DDA line algorithm.
2. Implementation of Bresenham's line algorithm
3. Implementation of midpoint circle algorithm
4. Implementation of midpoint ellipse algorithm
5. Implementation of Two Dimensional transformations - Translation, Rotation, Scaling, Reflection, Shear.
6. Implementation of composite 2D Transformations.
7. Cohen Sutherland 2D line clipping and Windowing.
8. Sutherland — Hodgeman Polygon clipping Algorithm.
9. Three dimensional transformations – Translation, Rotation, Scaling
10. Composite 3D transformations.
11. Drawing three dimensional objects and Scenes.
12. Generating Fractal images

BCS2601 Core and Advance Java

Course Objective:

1. Learn the Java programming language: its syntax, idioms, patterns, and styles.
2. Become comfortable with object oriented programming: Learn to think in objects

3. Learn the essentials of the Java class library, and learn how to learn about other parts of the library when you need them.
4. Introduce event driven Graphical User Interface (GUI) programming
5. How to develop multithreaded and networking applications and how to create dynamic pages.

Learning Outcome:

At the end of the course, the student should be able to:

1. Understand the basic concepts
2. Understand the basic concepts and principles of object oriented programming.
3. Understand the concepts of swing.
4. Learn to create JSP pages.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**INTRODUCTION TO CORE JAVA PROGRAMMING** History and evolution, Features of java environment**,** Difference from c and c+ +**,** The java architecture**,** Java Development Kit**,** Types of java programs**,** A sample java program**,** Compilation and Execution, Variable Declaration and arrays, Data types in java, Java Tokens, Variable Declaration, Type casting and conversion, Arrays, Operators in java, Operators-Introduction, Operator Precedence, Control Statements, Introduction to classes, Instance Variables, Class Variables, Instances Methods, Constructors, Declaring Object, Garbage collection, Method Overloading, Constructor Overloading, this reference, Using objects in methods, Recursion, Access modifiers, Inner class.	30 Hours	1
II	**INHERITANCE , ABSTRACT CLASSES AND INERFACES , EXCEPTION HANDLING AND MULTITHRAEDED PROGRAMMING** Inheritance**,** Super class variables and subclass Objects**,** The super reference, Constructor chaining**,** Method overriding**,** The final keyword**,** Abstract Classes and Interfaces**,** The abstract classes**,** The abstract methods, Defining interface, Implementing interface, Extending interface, Interface References ,Exception handling, Hierarchy of exception classes, Types of exception, Exception classes, Uncaught exceptions, Handling Exception, User defined Exception ,Multithreaded Programming, The java Thread model, The runnable interface, The thread class, Thread creation, Thread's Life cycle, Thread scheduling, Synchronization and Deadlock, Inter-	30 Hours	1

	thread communication, Joining threads, Suspending, resuming and stopping threads		
III	**PACKAGES AND ACCESS MODIFIERS, HANDLING STRING,INPUT OUTPUT CLASSES** Packages and access modifier, Recommended package naming convention, The Package Declaration, The CLASSPATH variable, The import statement, The java language packages , Importance of Package Design , Access Protection, Handling strings, Create Strings, Operations on strings, Character Extraction method, String comparison method, Searching and modifying strings, Data conversions and ValueOf( ) methods, Changing case of characters, The StringBuffer classes, Wrapper classes, Input and output classes, Hierarchy of classes in java.io package, File classes, InputStream and OutputStream classes , FilterInputStream and FilterOutputStreamClasses, Reader and writer classes, Basics of networking , Java's Networking protocol Hierarchy of classes in java.net package, Connection oriented protocol classes, Applet Life Cycle, Running Applets, Methods of Applet class, Graphics class, color class, font class, Limitation of applets, Inerfaces of java.applet package	30 Hours	1
IV	**AWT,LAYOUT MANAGEMENT AND EVENT HANDLING,SWING,JDBC,RMI,SERVLETS** AWT Classes, Hierarchy of classes in java.awt package, Component Classes, Container Classes, Frame Window in an Applet, MenusLayout management and event handling , Standard layout managers, Handling events, Event classes , Event Listener interface , Adapter classes, The Swing and the AWT, Swing packages, Structure of swing application, Top-Level swing containers, Lightweight swing container, JComponent Classes, Basic swing components, Swing Text Components, JDBC architecture, JDBC-ODBC-Relationship, Types of JDBC drivers, JDBC component, JDBC Interfaces and classes, Servlets, The Basic servlets architecture, Servlet designing , A Servlet that generates HTML, A Servlet to handle GET request, Java server pages, Introduction to net-beans	30 Hours	1

Text/Reference Books:

1. Patrick Naughton and Herbertz Schildt, "Java-2: The Complete Reference", TMH, 1999.
2. Bill Vanners, "Inside Java Virtual Machine", TMH, 2nd Ed.
3. Rick Dranell, "HTML 4 unleashed", Techmedia Publication, 2000
4. Shelley Powers, "Dynamic Web Publishing", 2nd Ed., Techmedia, 1998.
5. Paul Dietel and Harvey Deitel, "Java How to Program", PHI, 8th Ed., 2010
6. E. Balaguruswamy, "Programming with Java: A Primer", TMH, 1998.

BCS2602 Design & Analysis of Algorithm

Course Objective:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations

Learning Outcome:

At the end of the course, the student should be able to:

1. Argue the correctness of algorithms using inductive proofs and invariants.
2. Analyze worst-case running times of algorithms using asymptotic analysis.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them.
5. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms, and analyze them.
6. Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.
7. Explain the different ways to analyze randomized algorithms (expected running time, probability of error). Recite algorithms that employ randomization. Explain the difference between a randomized algorithm and an algorithm with probabilistic inputs.
8. Analyze randomized algorithms. Employ indicator random variables and linearity of expectation to perform the analyses. Recite analyses of algorithms that employ this method of analysis.
9. Explain what amortized running time is and what it is good for. Describe the different methods of amortized analysis (aggregate analysis, accounting, potential method). Perform amortized analysis.
10. Explain what competitive analysis is and to which situations it applies. Perform competitive analysis.
11. Compare between different data structures. Pick an appropriate data structure for a design situation.
12. Explain what an approximation algorithm is, and the benefit of using approximation algorithms.

Course Contents:

Module	Course Topics	Total	Credits

		Hours	
I	**Introduction and Advanced Data Structure** Notion of Algorithm, Analysis of algorithms, Designing of Algorithms, Growth of Functions, Master's Theorem Asymptotic Notations and Basic Efficiency Classes, Shorting and Searching Algorithm: Insertion Sort Selection Sort and Bubble Sort Divide and conquer - Merge sort , Quick Sort, Heap Sort, Sequential Search and Binary Search Binary Search Tree and Red Black tree: Traversal and Related Properties, Binomial Heaps, Fibonacci Heaps; Data Stricture for Disjoint Sets	30 Hours	1
II	**Advanced Design and Analysis Techniques** Dynamic Programming: Matrix chain multiplication Problem, Optimal Binary search tree etc., Greedy Algorithms; Amortized Analysis	30 Hours	1
III	**Graph Algorithms** Graph Algorithms: Elementary Graphs Algorithms, Depth first Search and Breadth First Search, Minimum Spanning Trees, Shortest paths problem Single-source Shortest Paths, All-Pairs Shortest Paths Maximum Flow and Flow networks, Back Tracking	30 Hours	1
IV	**Selected Topics** Randomized Algorithms, String Matching, Travelling Salesman Problem, NP Completeness , Approximation Algorithms	30 Hours	1

Text/Reference Books:

1. Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, Introduction to Algorithms, Second Edition, MIT Press/McGraw-Hill, 2001.
2. Jon Kleinberg and ÉvaTardos, Algorithm Design, Pearson, 2005.
3. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Wiley, 2006

BCS2603 Advanced Computer Architecture

Course Objective:

1. Provide in-depth coverage of current and emerging trends in computer architectures, focusing on performance and the hardware/software interface.
2. The course emphasis is on analyzing fundamental issues in architecture design and their impact on application performance.

Learning Outcome:

At the end of the course, the student should be able to:

1. Understand advanced issues in design of computer processors, caches, and memory.
2. Analyze performance trade-offs in computer design.
3. Apply knowledge of processor design to improve performance in algorithms and software systems.

4. Acquire experience with tools for statistical analysis of instruction set trade-offs.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction** Introduction Parallel Computing, Parallel Computer Model, Program and Network Properties, Parallel Architectural Classification Schemes. Flynn's & Feng's Classification. Performance Metrics and Measures, Multiprocessor System and Interconnection Networks. Speedup Performance Laws.	30 Hours	1
II	**Pipelining and Memory Hierarchy** Basic and Intermediate Concepts: Instruction Set Principle, ILP: Basics, Exploiting ILP, Limits on ILP. Linear and Nonlinear Pipeline Processors, Super Scalar and Super Pipeline Design. Memory Hierarchy Design, Advanced Optimization of Cache Performance, Memory Technology and Optimization, Cache Coherence and Synchronization Mechanisms.	30 Hours	1
III	**Thread and Process Level Parallel Architecture** Introduction to MIMD Architecture, Multithreaded Architectures, Clustering, Instruction Level Data Parallel Architecture. SIMD Architecture, Associative and Neural Architecture, Data Parallel Pipelined, Systolic Architectures, Vector Architectures, Parallel Algorithms: Parallel Reduction, Prefix Sums, Pre-order Tree Traversal, Merging two Sorted lists; Matrix Multiplication: Row Column Oriented Algorithms, Parallel Algorithm Design Strategies.	30 Hours	1

Text/Reference Books:

1. Kai Hwang," Advance Computer Architecture", TMH
2. Matthew, "Beginning Linux Programming", SPD/WROX
3. Hennessy and Patterson," Computer Architecture: A Quantitative Approach", Elsevier
4. Dezso and Sima, "Advanced Computer Architecture", Pearson
5. Quinn, "Parallel Computing: Theory & Practice", TMH
6. Quinn, "Parallel Programming in C with MPI and Open MP", TMH
7. Open MP Specification and Usage (www.openmp.org)

BCS2604 Compiler Design

Course Objective:

1. Understand the principles governing all phases of the compilation process.
2. Understand the role of each of the basic components of a standard compiler.
3. Have awareness of the problems of and methods and techniques applied to each phase of the compilation process.
4. Apply standard techniques to solve basic problems that arise in compiler construction.
5. Understand how the compiler can take advantage of particular processor characteristics to generate good code.

Learning Outcome:

At the end of the course, the student should be able to:

1. To understand, design and implement a lexical analyzer.
2. To understand, design and implement a parser.
3. To understand, design code generation schemes.
4. To understand optimization of codes and runtime environment.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction to Compiler** Introduction-Compiler and features, Analysis – Synthesis model, Phases of compiler, Grouping of Phase, Concept of pass, Bootstrapping. Lexical analyser-Implementation and Role of lexical analyser, Token, Pattern and Lexemes and function, Finite state machine and Regular expression. Formal grammar and their application to syntax analysis-CFG, derivation and Parse trees **Basics Parsing Techniques –I**: **Top-Down Parser**-Back tracking, Predictive Parser	30 Hours	1
II	**Basic parsing techniques-II: Bottom –Up Parser** Shift –Reduce Parser, LR Parser-LR(0) Parser , SLR parser , Canonical LR Parser, LALR Parser. Syntax-Directed Translation-Syntax Directed definition and type, Construction of Syntax Tree and DAG representation. Symbols Table- Storage Allocation strategies, Runtime Storage Organization, Structure Storage Allocation, Error detection and Recovery-Lexical, syntax and semantic error.	30 Hours	1
III	**Intermediate code Generator** Intermediate code- forms of intermediate code, Implementation of 3-address code- Quadruple, direct and triple, Translation of Assignment statements, Translation of Boolean expression- Flow of Control Statement, and Case statements. Code Generation-	30 Hours	1

	Design issues, The Target code and addresses in target code, Code generator algorithm.		
IV	**Code Optimization** Machine-independent optimization, Basic Blocks, Flow graph- DAG representation of basic blocks, Loop in Flow graph .Transformation, Loop Optimization ,Peephole Optimization, Global Data Flow-Control Flow Analysis, Data Flow Analysis	30 Hours	1

Text/Reference Books:

1. Aho, Sethi& Ullman, "Compilers: Principles, Techniques and Tools", Pearson Education
2. V Raghvan, "Principles of Compiler Design", TMH
3. Kenneth Louden, " Compiler Construction", Cengage Learning.
4. 4.. Charles Fischer and Ricard LeBlanc," Crafting a Compiler with C", Pearson Education

Generic Elective-I

BCS2011 Multimedia Systems

Course Objective:

1. The aim of the course is that to acquaint a student with number concepts about the signal processing exploited in the field of multimedia applications.
2. Some basic tools, such as Fourier and Cosine transforms, are presented.
3. Nyquist's theorem and a short introduction to digital filter design and usage.
4. The general notion of waveform is introduced and discussed along with a short presentation of typical filter arrangements: low-band, high-band, pass-band, etc...
5. Different color models are also discussed and the concepts of Chrominance, Luma and Luminance are introduced within the general mechanism of human vision system.
6. The most popular compression algorithms are widely discussed as well as audio systems such as MIDI, MPEG and MPEG4.
7. The JPEG compression is presented as an instance of the so called Discrete Cosine Transform even if the complete compression cycle is built on the entropy-based compression algorithms.

Learning Outcome:

By studying this course, students will be able to

1. Be familiar with multimedia data types and the conversion between analogue and digital forms.
2. Understand the characteristics of different media; understand the representations of different multimedia data; understand different data formats; be able to take into considerations in multimedia system designs;
3. Understand the characteristics of human's visual system; understand the characteristics of human's audio system; be able to take into considerations in multimedia techniques design and implementation;
4. Understand different compression principles; understand different compression techniques; understand different multimedia compression standards; be able to design and develop multimedia systems according to the requirements of multimedia applications.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction** Introduction: What is multimedia? Defining the scope of multimedia. Applications of multimedia, hardware and software requirements, multimedia database.	30 Hours	1
II	**Digital representation** Digital representation: Introduction, Analog representation, waves, digital representation, need for digital representation, A to D conversion, D to A conversion, relation between sampling rate and bit depth, Quantization error, Fourier representation,	30 Hours	1

	pulse modulation. Importance and drawback of digital representation.		
III	**Text ,Image, Audio and Video** Text and Image: Introduction, Types of text, Font, insertion, compression, File formats. Types of images, color models, Basic steps for image processing, principle and working of scanner and digital camera, Gamma and gamma correction. Audio and Video technology: Fundamental characteristics of sound, psycho- acoustics, Raster scanning principles, sensors for TV cameras, color fundamentals, additive and subtractive color mixing, Liquid crystal display (LCD), Plasma Display, Panel (PDP), file formats	30 Hours	1
IV	**Compression and use of authoring tools** Compression and coding: What is compression? Need for compression, Types of compression- basic compression techniques-run length, Huffman's coding, JPEG, zip coding. Overview of Image and Video compression techniques. Multimedia presentation and authoring: Overview, multimedia authoring metaphor, multimedia production, presentation and automatic authoring, Design paradigms and user interface, Overview of tools like Adobe Premier, Director, Flash and Dreamweaver.	30 Hours	1

Text/Reference Books:

1. Data Compression – David Salomon , Springer Publication, 4th Edition.
2. Introduction to Data Compression – Khalid Sayood, Morgan Kaufmann Series, 3rd Edition

BCS2012 Soft Computing

Course Objective:

1. Learn the basic concept of fuzzy set theory.
2. Understand the working principle of various AI techniques and heuristic search algorithms.
3. Learn about the architecture of artificial neural networks and implement them in fuzzy environment.
4. Study the concept behind genetic algorithm and its various operations.
5. Learn different levels of CPN Networks and ART algorithms.

Learning Outcome:

At the end of the course, the student should be able to:

1. Implement numerical methods in soft computing.
2. Design the algorithms which can work as an intelligent production system.

3. Demonstrate various learning methods in artificial neural networks, like: supervised and unsupervised learning.
4. Familiar about the various mutation and cross over techniques of genetic algorithm for producing new strings.
5. Design and Implement different predicate logic rules for solving any specific AI problem.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Artificial Intelligence** Soft Computing; Introduction of soft computing, soft computing vs. hard computing, various types of soft computing techniques; Artificial neural networks, Fuzzy logic, Genetic Algorithms. Applications of soft computing. Artificial Intelligence; Introduction of AI. Production systems; Types of production systems, Characteristics of production systems. Search Algorithms; Breadth first search, Depth first search. Heuristic Search Algorithms; Hill Climbing, Best first Search, A* algorithm, AO* Algorithms. Knowledge representation issues. Prepositional and predicate logic. Forward Reasoning and backward reasoning. Weak & Strong Slot & filler structures. Natural Language Processing (NLP).	30 Hours	1
II	**Neural networks** Structure of Biological neuron; Neuron, Nerve Structure and synapse. Artificial Neuron and its model. Activation functions. Neural network architecture; Single layer feed forward networks, multilayer feed forward networks, Recurrent networks. Various learning techniques; Perceptron training algorithm; Linear separability, Widrow & Hebb's learning rule/Delta rule. ADALINE v/s MADALINE. Introduction of MLP and BPN. Error back propagation algorithm (EBPA); Characteristics and application of EBPA, momentum factor and limitation of EBPA. Difference between ANN and human brain. Characteristics and applications of ANN. Associative Memory and its characteristics. Counter propagation network (CPN); Architecture of CPN, functioning & characteristics of CPN. Hopfield/ Recurrent network. Hopfield v/s Boltzman machine. Adaptive Resonance Theory (ART); Architecture of ART, Classification and training of ART.	30 Hours	1
III	**Fuzzy logic** Basic concepts of fuzzy logic. Fuzzy sets versus Crisp sets. Fuzzy set theory and operations. Properties of fuzzy sets and crisp sets. Fuzzy relations and Crisp relations. Fuzzy to Crisp conversion. Membership	30 Hours	1

	functions. Fuzzyfications & Defuzzifications. Fuzzy preposition. Fuzzy inference System. Fuzzy Rule Base. Fuzzy reasoning and decision making. Fuzzy Logic Controller (FLC). Formation, decomposition & aggregation of fuzzy rules. Industrial applications of Fuzzy.		
IV	**Genetic Algorithm(GA)** Introduction of Genetic Algorithm; Fundamentals of GA, Basic concepts of GA, Working principle, Encoding, fitness function, Reproduction. Genetic modelling; Inheritance operator, Cross over Operators, Inversion & deletion, Mutation operator, Bitwise operators. Generational Cycle of GA. Convergence of GA. Applications & advances in GA. Differences& similarities between GA & other traditional methods.	30 Hours	1

Text/Reference Books:

1. S. Rajsekaran & G.A. Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications" Prentice Hall of India.
2. Sivanandani, Deepa, "Principles of Soft Computing", Wiley India (2007)
3. Jang J.S.R. Sun C.T. and Mizutani E., "Neuro-Fuzzy and Soft computing", Prentice Hall
4. Timothy J. Ross. "Fuzzy Logic with Engineering Applications", McGraw Hill
5. Laurene Fausett. 'Fundamentals of Neural Networks", Prentice Hall
6. D.E. Goldbery, "Genetic Algorithms: Search, Optimization and Machine Learning". Addison Wesley.

BCS2013 Evolutionary Algorithms

Course Objective:

1. How to solve hard problems without using complex mathematical formulations
2. Design algorithms that are robust yet easy to program
3. To solve optimization related problems efficiently.

Learning Outcome:

Upon completion of this course, students will be able to:

1. Explain the principles underlying Evolutionary Computation in general and Genetic Algorithms in particular.
2. Apply Evolutionary Computation Methods to find solutions to complex problems
3. Analyze and experiment with parameter choices in the use of Evolutionary Computation
4. Summarize current research in Genetic Algorithms and Evolutionary Computing

Course Contents:

Module	Course Topics	Total	Credits

		Hours	
I	Introduction to Evolutionary Computation, Biological Background: Principles of Darwinian natural selection, Historical Development of EC, Genetic Algorithms , Genetic Programming, Evolutionary Strategies and Evolutionary Programming , Features of Evolutionary Computation, Advantages of Evolutionary Computation Applications of Evolutionary Computation.	30 Hours	1
II	Genetic Algorithms: Overview of Conventional Optimization and Search Techniques, Simple Genetic Algorithm, terminology: Individual, Genes, Fitness, Population,, Encoding, Breeding,, Termination, Comparison with Other Optimization , Techniques GA in search, optimization, and machine learning. Case Study of Travelling Salesman Problem	30 Hours	1
III	Evolutionary Strategies: Introduction, Comparison with GA &GP Operators : Gaussian Mutation Operator and Intermediate Recombination Operator. Application of ES for Image Enhancement	30 Hours	1
IV	Foundations of Evolutionary Algorithms , Schemas and the two-armed bandit problem, Advantages and disadvantages of evolutionary algorithms over alternative methods. Co-evolutionary Algorithms: Cooperative co-evolution, Competitive co-evolution, Swarm intelligence and ant colony optimization.	30 Hours	1

Text/Reference Books:

1. Sivanandam, Deepa "Introduction to Genetic Algorithm", Springer.
2. Melanie Mitchell: "An Introduction to Genetic Algorithm", Prentice Hall of India.
3. D. E. Goldberg, "Genetic Algorithms in Search, Optimisation and Machine Learning", Addison-Wesley.
4. Zbigniew Michalewics, "Genetic Algorithms + Data Structures = Evolution Programs", Springer Verlag, 1997.
5. Goldberg, "Genetic Algorithms", Pearson Eduction.
6. T. Back, D. B. Fogel and Michalewicz, "Evolutionary Computation1: Basic Algorithms and Operators", 2000.

BCS2014 Distributed Database System

Course Objective:

1. Learn to manage large volume of shared data
2. To understand fundamental concepts in a parallel environment.
3. To implement large database in a distributed environment.
4. To provide insight into related research problems.
5. Techniques to learn distributed database design.

Learning Outcome:

At the end of the course, the student should be able to:

1. Explain the techniques used for data fragmentation, replication, and allocation during the distributed database design.
2. Evaluate simple strategies for executing a distributed query to select the strategy that minimizes the amount of data transfer.
3. Explain how the two phase commit protocol is used to deal with committing a transaction that accesses databases stored on multiple nodes.
4. Describe distributed concurrency control based on the distinguished copy techniques and the voting methods.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	INTRODUCTION: Distributed data processing, What is a DDBS?, Advantages and disadvantages of DDBS, Problem areas, Overview of database and computer network concepts DISTRIBUTED DATABASE MANAGEMENT SYSTE ARCHITECTURE: Transparencies in a distributed DBMS, Distributed DBMS architecture, Global directory issues	30 Hours	1
II	DISTRIBUTED DATABASE DESIGN: Alternative design strategies, Distributed design issues, Fragmentation, Data allocation TRANSACTION MANAGEMENT: The transaction concept, Goals of transaction management, Characteristics of transactions, Taxonomy of transaction models CONCURRENCY CONTROL: Concurrency control in centralized database systems, Concurrency control in DDBSs, Distributed concurrency control algorithms, Deadlock management	30Hours	1
III	SEMANTICS DATA CONTROL: View management, Data security, Semantic Integrity Control QUERY PROCESSING ISSUES: Objectives of query processing, Characterization of query processors, Layers of query processing, Query decomposition, Localization of distributed data DISTRIBUTED QUERY OPTIMIZATION: Factors governing query optimization, Centralized query optimization, Ordering of fragment queries, Distributed query optimization algorithms	30 Hours	1
IV	RELIABILITY: Reliability issues in DDBSs, Types of failures, Reliability techniques, Commit protocols, Recovery protocols PARALLEL DATABASE SYSTEMS: Parallel architectures, Parallel query processing and optimization, Load balancing ADVANCED TOPICS : Mobile Databases,	30 Hours	1

	Distributed Object Management, Multi-databases		

Text/Reference Books:

1. Principles of Distributed Database Systems, M.T. Ozsu and P. Valduriez, Prentice-Hall, 1991. (Required).
2. Distributed Database Systems, D. Bell and J. Grimson, Addison-Wesley, 1992. (Referenced).
3. Distributed Database Management Systems: A Practical Approach by Saeed K. Rahimi (Author), Frank S. Haug.

BCS2651 Java Lab

1. Programs on class, constructor
2. Programs on overloading, inheritance, overriding
3. Programs on wrapper class, vectors, arrays
4. Programs on developing interfaces- multiple inheritance, extending interfaces
5. Programs on creating and accessing packages
6. Programs on multithreaded programming,
7. Programs on handling errors and exceptions
8. Applet programming and graphics programming

BCS2652 Design & Analysis of Algorithm Lab

Programming assignments on each algorithmic strategy:

1. Divide and conquer method (quick sort, merge sort, Strassen's matrix multiplication),
2. Greedy method (knapsack problem, job sequencing, optimal merge patterns, minimal spanning trees).
3. Dynamic programming (multistage graphs, OBST, 0/1 knapsack, traveling salesperson problem).
4. Back tracking (n-queens problem, graph coloring problem, Hamiltonian cycles).
5. Sorting : Insertion sort, Heap sort, Bubble sort
6. Searching : Sequential and Binary Search
7. Selection : Minimum/ Maximum, K-th smallest element

BCS2701 Web Technology

Course Objective:

1. Discuss how XML DTDs differ from XML schemas. Recognize and correct validity errors in XML documents. Write documents that make appropriate use of XML namespaces.
2. Write client-server applications that communicate via XML documents. Discuss ways in which an XSL transform differs from processing an XML document using a DOM API.
3. Describe how a given web server responds to an HTTP request for a dynamic resource. Describe sessions conceptually and explain how the concept can be implemented using cookies and URL rewriting.
4. Discuss the concept and implementation of cookies as well as related privacy concerns.
5. Explain common security threats such as cross-site scripting and misformed HTTP requests and demonstrate avoidance techniques for each.

6. Use CSS to implement a variety of presentation effects in HTML and XML documents, including explicit positioning of elements
7. Demonstrate techniques for improving the accessibility of an HTML document
8. Write a valid standards-conformant HTML document involving a variety of element types, including hyperlinks, images, lists, tables, and forms
9. Describe the actions, including those related to the cache, performed by a browser in the process of visiting a Web address
10. Install a web server and perform basic administrative procedures, such as tuning communication parameters, denying access to certain domains, and interpreting an access log

Learning Outcome:

At the end of the course, the student should be able to:

1. Install, configure, and maintain a Web server;
2. Identify and successfully manage issues of Web caching, Web security, and Web performance;
3. proficient in Web technologies like HTML, XML, DOM, SAX
4. java technologies for the Web like JAXP, Java Beans, Servlet, JSP, JDBC with proficiency in Web server configuration
5. Create XML documents. Create XML Schema. Build dynamic web pages using JavaScript (client side programming).
6. Create web pages using XHTML and Cascading Styles sheets.
7. Analyze a web page and identify its elements and attributes.
8. Build web applications using JSP.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**HTML, XML and Java Scripting** HTML Common tags: List, Tables, images, forms, Frames; Cascading Style sheets, Introduction to Java Scripts, Objects in Java Script, Dynamic HTML with Java Script, XML, Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors, DOM and SAX and XSTL ,Java API for XML Processing (JAXP)	30 Hours	1
II	Java Beans: Introduction to Java Beans, Advantages and properties of Java Beans, BDK , Introspection, Using Bound properties, Bean Info Interface ,Constrained properties, Introduction to EJB, Java Beans API, Introduction to Servelet, Lifecycle of a Servelet, JSDK ,The Servelet API, The javax. servelet Package, The javax. servelet, HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues	30 Hours	1
III	Introduction to JSP,JSP Processing, JSP Application Design ,Tomcat Server ,Implicit JSP Objects, Conditional Processing, Declaring Variables and	30 Hours	1

	Methods, Error Handling and Debugging, Sharing Data Between JSP pages, Sharing Session and Application Data		
IV	Database Access, Database Programming using JDBC, Studying Javax.sql.* package, Accessing a Database from a JSP Page, Application – Specific Database Actions, Deploying JAVA Beans in a JSP Page, Introduction to struts framework.	30 Hours	1

Text/Reference Books:

1. Programming world wide web-Sebesta, Pearson
2. Java: the complete reference, 7th edition, Herbert Schildt, Tata McGraw-Hill.
3. Core Servelet and Java Server Pages Volume 1: Core Technologies by Marty Hall and Larry Brown Pearson
4. Java Server Pages, 2nd Edition by Hans Bergsten, Publisher: O'Reilly
5. Internet and World Wide Web – How to program by Dietel and Nieto Prentice Hall of India/Pearson Education Asia.

BCS2702 Network Security and Cryptography

Course Objective:

1. Have a fundamental understanding of the objectives of cryptography and network security
2. Become familiar with the cryptographic techniques that provide information and network security
3. To know the different types of algorithms of exchanging information in a secret way.
4. To know the possible threats which can breach the secure communication

Learning Outcome:

At the end of the course, the student should be able to:

1. Understand cryptography and network security concepts and applications
2. Apply security principals to system design
3. Identify and investigate network security threats
4. Analysis of network traffic and security threats

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction to Cryptography and Symmetric Ciphers** Security Attacks: Security Services and mechanism; Classical encryption techniques: Substitution ciphers and Transposition ciphers, Steganography, Cryptanalysis; Modern Block Ciphers: Stream and Block Cipher, Block Cipher Principles, Block Cipher Modes of Operations; Shannon's theory of Confusion and Diffusion; Fiestal structure; Data encryption standard(DES); Strength of DES; Idea of differential cryptanalysis; Triple DES; Symmetric Key	30 Hours	1

	Distribution; Finite Fileds: Introduction to groups, rings and fields, Modular Arithmetic, Euclidean Algorithm, Finite Fields of the form GF(p)		
II	**Basics of Number Theory and Public key Cryptography** Introduction to Number Theory: Prime and Relative Prime Numbers, Fermat's and Euler's theorem, Testing for Primality, Chinese Remainder theorem, Discrete Logarithms; Public Key Cryptography: Principles of Public-Key Cryptography, RSA Algorithm, Security of RSA; Key Management: Deffie-Hellman Key Exchange, ElGamal Public key cryptosystems.	30 Hours	1
III	**Hash Functions and Digital Signatures** Message Authentication; Hash Functions; Secure Hash Functions; Security of Hash functions and MACs; Digital Signatures; Digital Signature Standards (DSS); Proof of digital signature algorithm; Advanced Encryption Standard (AES) encryption and decryption.	30 Hours	1
IV	**Network and System Security** Authentication Applications: Kerberos, X.509 Certificates; Electronic Mail Security: Pretty Good Privacy, S/MIME; IP Security: IP Security Architecture, Authentication Header, Encapsulating security payloads, Combining Security Associations; Web Security: Secure Socket Layer and Transport Layer Security, Secure Electronic transaction; Intruder; Viruses; Firewalls.	30 Hours	1

Text/Reference Books:

1. William Stallings, "Cryptography and Network Security: Principals and Practice", Pearson Education.
2. Behrouz A. Frouzan: Cryptography and Network Security, Tata McGraw-Hill
3. Bruce Schiener, "Applied Cryptography". John Wiley & Sons
4. Bernard Menezes," Network Security and Cryptography", Cengage Learning.
5. AtulKahate, "Cryptography and Network Security", Tata McGraw-Hill

Generic Elective II

BCS2021 System Programming & System Administration

Course Objective:

1. To explain the basic operations that are performed from the time a computer is turned on until a user is able to execute programs.
2. To work effectively in a UNIX-style environment.

3. To provide you with a basic understanding of the issues involved in writing system programs, manipulating system processes, system I/O, system permissions, files, directories, signals, threads, sockets, terminal, etc.
4. To operate in and self-manage in programming teams.
5. Design and implement programs making direct use of operating system facilities to perform low- level file I/O and directory manipulations

Learning Outcomes:

At the end of the course, the student should be able to:

1. Access the UNIX system and perform basic operations, including using help features.
2. Access and manipulate files and directories, including basic and advanced directory and file
3. Use file system utilities.
4. Troubleshoot system processes.
5. Perform environment customization.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction to System Programming** Evolution of Components Systems Programming**,** Assemblers, Loaders, Linkers, Macros, Compilers. Software tools: Text editors, Interpreters and program generators, Debug Monitors; Compiler: Brief overview of compilation process, Incremental compiler; Assembler : single phase and two phase assembler, symbol table; Loader : Loader schemes, compile and go Loader, general loader schemes, absolute loader, Subroutine linkage, Reallocating loader, Direct linkage Loader, Binders, Linking loader; Macros :Macro language and macro-processor, macro instructions, features of macro facility, macro instruction arguments, conditional macro expansion, macro calls with macro instruction defining macros.	30 Hours	1
II	**Concept of Unix Operating System** Basic Features Of Operating System**,** CPU scheduling; Memory Management: Swapping, and demand paging; File system**;** Getting Started with Unix**:** User names and groups, logging in**,** Format of Unix commands**,** Changing your password;**,** Characters0with special meaning**;** Unix documentation.	30 Hours	1
III	**Files and directories:** Directories**,** Current directory, looking at the directory contents**,** Absolute and relative pathnames, Some Unix directories and files; Files: Looking at the file contents; File permissions, Basic operation on files, changing permission modes; Standard files, standard output; Standard input, Standard error; filters and pipelines. Processes: Finding out about processes; Stopping	30 Hours	1

	background process; Unix editor vi, Test Manipulation, Inspecting files, File statistics, Searching for patterns, Comparing files, Operating on files, Printing files, Rearranging files, Sorting files, Splitting files, Translating characters, AWK utility.		
IV	**Shell programming and System Administration:** Shell programming, Programming in the Borne and C-Shell, Wild cards, Simple shell programs, Shell variables, Shell programming constructs, Interactive shell scripts, Advanced features; System Administration: Definition of system administration, Booting the system Maintaining user accounts, File systems and special files, Backups and restoration, Role and functions of a system manager; Overview of the Linux operating system.	30 Hours	1

Text/Reference Books:

1. Systems Programming by Donovan, Tata McGraw-Hill.
2. The unix programming environment by Brain Kernighen& Rob Pike, 1984, Prentice Hall of India & Rob Pike.
3. Design of the Unix operating system by Maurich Bach, 1986, Prentice Hall of India.
4. Introduction to UNIX and LINUX by John Muster, 2003, Tata McGraw-Hill.
5. Advanced Unix programmer's Guide by Stephen Prato, BPB
6. Unix- Concept and applications by Sumitabha Das, 2002, T.M..H

BCS2022 System Modeling & Simulation

Course Objective:

1. The basic system concept and definitions of system.
2. Techniques to model and to simulate various systems the ability to analyze a system and to make use of the information to improve the performance.

Learning Outcome:

At the end of the course, the student should be able to:

1. Define basic concepts in modelling and simulation (M&S)
2. Classify various simulation models and give practical examples for each category
3. Construct a model for a given set of data and motivate its validity
4. Generate and test random number variants and apply them to develop simulation models
5. Analyze output data produced by a model and test validity of the model
6. Explain parallel and distributed simulation methods

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**INTRODUCTION TO SIMULATION** Simulation: Simulation as a tool, Advantages and Disadvantages of Simulation, Areas of Application, Systems and System Environment. Components of a	30 Hours	1

	System. Discrete and Continuous Systems. Model of a System, Types of Models, Discrete-Event System Simulation. Steps of Simulation Study		
II	**GENERAL PRINCIPLES** Concepts in Discrete-Event Simulation: The Event-Scheduling / Time-Advance: Algorithm, World Views. Manual simulation, Using Event Scheduling. Properties of Random Numbers, Generation of Pseudo-Random Numbers. Techniques for Generating Random Numbers. Tests for Random Numbers	30 Hours	1
III	**RANDOM-VARIATE GENERATION** Inverse Transform technique: Exponential Distribution, Uniform Distribution, Discrete Distributions, Acceptance-Rejection Technique, Poisson Distribution. Data Collection, Identifying the distribution with Data, Parameter Estimation, Goodness of Fit Tests, Selecting Input Models without Data Multivariate and Time-Series Input Models.	30 Hours	1
IV	**VERIFICATION AND VALIDATION OF SIMULATION MODELS** Model Building, Verification and Validation: Verification of Simulation Models, Calibration and Validation of Models. Types of Simulations with Respect to Output Analysis .Stochastic Nature of Output Data. Measures of Performance and Their Estimation. Output Analysis for Terminating Simulations, Output Analysis for Steady-State Simulations. Simulation Tools, Model Input. High-Level Computer-System Simulation, CPU Simulation, Memory Simulation.	30 Hours	1

Text/Reference Books:

1. Jerry Banks, John S. Carson, Barry L. Nelson, David M. Nicol, "Discrete-Event System Simulation", Third Edition, Prentice-Hall India
2. Averill M. Law, W. David Kelton, "Simulation Modelling and Analysis" Third Edition, McGraw Hill.
3. Geoffrey Gordon, "System Simulation", Second Edition, Prentice-Hall India.

BCS2023 Artificial Intelligence

Course Objective:

1. To have an understanding of the achievements of AI.
2. To have an appreciation for the engineering issues underlying the design of AI systems.
3. To have a basic proficiency in a traditional AI language
4. To have an understanding of the basic issues of knowledge representation
5. For understanding the concept of blind and heuristic search,

6. Understanding of minimax and resolution concept of AI.

Learning Outcome:

At the end of the course, the student should be able to:

1. Write simple to intermediate programs written in LISP/PROLOG.
2. To understand code written in above language.
3. To implement AI topics such as learning, natural language processing.
4. Understand and Design agents and robotics, expert systems.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	Introduction to Artificial Intelligence, Simulation of Sophisticated & Intelligent Behaviour, Problem solving in games, Natural language, Automated reasoning , Visual Perception, Heuristic algorithm versus solution guaranteed algorithms.	30 Hours	1
II	First order predicate calculus, Horn Clauses, Semantic Nets, Partitioned Nets, Minskey frames Knowledge Sources Expert, Knowledge Acquisition, Knowledge Representation, Knowledge Base, Production Rules Knowledge Base, Interference Strategies, Forward and Backward Chaining	30 Hours	1
III	Expert System : Existing Systems (DENDRAL, MYCIN), Domain exploration, Meta Knowledge, Expertise Transfer, Self Explaining System	30 Hours	1
IV	Introduction to pattern Recognition, Structured Description, Symbolic Description, Machine perception, Line Finding, Interception, Semantic Model Object Identification, Speech Recognition, Programming Language: Introduction to programming Language LISP, Introduction to programming Language PROLOG	30 Hours	1

Text/Reference Books:

1. Artificial Intelligence: A Modern Approach, 2nd edition, by Russell and Norvig, Prentice Hall.
2. LISPcraft by Robert Wilensky, W.W. Norton. Software Allegro Common Lisp.
3. Charnick “Introduction to Artificial Intelligence.” Addision Wesley
4. Rich & Knight, “Artificial Intelligence”.
5. Winston, “LISP”, Addison Wesley

BCS2024 Embedded System Design

Course Objective:

1. To impart fundamental concepts in the area of Embedded Systems.
2. To impart the design an embedded system.
3. To impart the partition a system to hardware and software parts efficiently.
4. To impart the Hardware/software Co-design concepts.

Learning Outcome:

At the end of the course, the student should be able to:

1. Design embedded system architectures for various applications.
2. Identify, formulate, and solve engineering problems
3. Learn Function on multidisciplinary teams.
4. Apply knowledge of mathematics, science and engineering.
5. Explain applications, benefits, and limitations of networked embedded systems for environmental science, health, and safety, industrial, and consumer usage objectives.
6. Develop standard project plans for a software development team including interface definition.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction** Definition, Embedded System Project Management, Embedded System Design and Issues, Design Cycle, Use of Target System and In-Circuit Emulator, Use of Software Tools for Development of Embedded System	30 Hours	1
II	**RTOS and Microcontroller** Task and Task States, Task and Data, Semaphores and Shared Data Operating System Services: Message Queues, Timer Functions, Events, Memory management, Interrupts Routines in an RTOS, Microprocessor Vs. microcontroller, 8051 microcontroller	30 Hours	1
III	**Embedded System Development** Embedded System Evolution Trends, Round Robin, Robin with Interrupts, Function One Scheduling Architecture, Assembler, Compiler, Cross Compiler and IDE, Object Oriented Interfacing, Recursion, Debugging Strategies, Simulators	30 Hours	1
IV	**Networks for Embedded System** I2C Bus, CAN Bus, SHARC Link Ports, Ethernet, Myrinet, Internet, Bluetooth, IEEE 1149.1 (JTAG) Testability	30 Hours	1

Text/Reference Books:

1. "Embedded Systems", Raj Kamal,TMH.
2. "The 8051 Microcontroller", K. J. Ayala, Penram International.
3. "Design with PIC Icrocontroller", J. b. Peatman, Printice Hall.
4. "Real Time Systems", H.Kopetz, Kluwer, 1997.
5. "Co-synthesis of hardware and software for embedded systems", R. Gupta, Kluwer, 1996.

Generic Elective – III

BCS2031 Cyber Law and Security

Course Objective:

1. Teach students the basics of information security and computer communication.
2. Familiarize students with Cyber Laws and Security policies and Cryptography.
3. Learn some of the convention and experience the ways of improving from existing experiences.

Learning Outcome:

At the end of the course, the student should be able to:

1. They can help the organization to continue its commercial activities in the event of significant information security incidents
2. Students can establish responsibility and accountability for information security in organizations
3. To be proficient in various forensic tools and usage of tools for disk imaging and recovery processes
4. The students will be able to design security procedures and policies
5. They can be well versed in various security standards and security testing techniques

Course Contents:

Module	**Course Topics**	**Total Hours**	**Credits**
I	Information Systems and its Importance**,**History of Information Systems and its basics, Changing Nature of Information Systems, Need of Distributed Information Systems, Role of Internet and Web Services, Information System Threats and attacks, Classification of Threats and Assessing Damages Security in Mobile and Wireless Computing , Security Challenges in Mobile Devices, authentication Service Security, Security Implication for organizations, Laptops Security Concepts in Internet and World Wide Web: Brief review of Internet Protocols-TCP/IP, IPV4, IPV6, Functions of various networking components routers, bridges, switches, hub, gateway and Modulation Techniques.	30 Hours	1
II	Basic Principles of Information Security, Confidentiality, Integrity Availability and other terms in Information Security, Information Classification and their Roles, Security Threats to E Commerce, Virtual Organization, Business Transactions on Web, E Governance and EDI, Concepts in Electronics payment systems, E Cash, Credit/Debit Cards, Physical Security- Needs, Disaster and Controls, Basic Tenets of Physical Security and Physical Entry Controls, Access Control- Biometrics, Factors in Biometrics Systems, Benefits ,Criteria for selection of biometrics, Design Issues in Biometric Systems ,Interoperability Issues, Economic and Social	30 Hours	1

	Aspects, Legal. Challenges Framework for Information Security, ISO 27001, SEE-CMM, Security Metrics, Information Security Vs. Privacy		
III	Cryptographic Systems, Model of Cryptographic Systems, Issues in Documents Security, System of Keys, Public Key Cryptography, Digital Signature, Requirement of Digital Signature System, Finger Prints, Firewalls Design and Implementation Issues, Policies Network Security, Basic Concepts, Dimensions, Perimeter for Network Protection, Network Attacks, Overview of IDS, Need of Intrusion Monitoring and Detection, Intrusion Detection Virtual Private Networks- Need, Use of Tunneling with VPN, Authentication Mechanisms, VPN, Types of VPNs and their Usage, Security Concerns in VPN.	30 Hours	1
IV	Laws, Investigation and Ethics, Cyber Crime , Information Security and Law ,Types & overview of Cyber Crimes**,**Cyber Law, Issues in E-Business Management Overview of Indian IT Act**,**Ethical Issues in Intellectual property rights**,** Copy Right, Patents, Data privacy and protection**,**Domain Name, Software piracy, Plagiarism, Issues in ethical hacking.	30 Hours	1

Text/Reference Books:

1. Godbole,“ Information Systems Security”, Willey
2. Merkov, Breithaupt,“ Information Security”, Pearson Education
3. Yadav, “Foundations of Information Technology”, New Age, Delhi
4. Schou, Shoemaker, “ Information Assurance for the Enterprise”, Tata McGraw Hill
5. Sood,“ Cyber Laws Simplified”, McGraw Hill
6. Furnell, “Computer Insecurity”, Springer
7. IT Act 2000

BCS2032 Real Time Operating System

Course Objective:

1. This course will provide students with an introduction to operating systems theory
2. Practical problem solving approaches to real-time systems.
3. Real-time scheduling and schedulability analysis
4. Formal specification and verification of timing constraints and properties
5. Design methods for real-time systems

Learning Outcome:

At the end of the course, the student should be able to:

1. List characteristics of real-time operating systems (RTOS).
2. Compare hard and soft real-time systems.
3. List features and services that are typically provided by an RTOS.
4. Configure the scheduler that is used in an RTOS.

5. Write applications that create and delete tasks, control task scheduling, and obtain task information.
6. Compare binary semaphores, counting semaphores, and mutexes.
7. Describe how semaphores are typically used in RTOS applications.
8. Write applications that create, delete, acquire and release a semaphore.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	Definition of Real Time Operating Systems, Classification of RTOS, Concept of computer control, Sequence, loop and supervisor control, Centralized, hierarchical and distributed systems, Human Computer interface , Hardware requirement for real time applications, Specialized processors, interfaces and communications	30 Hours	1
II	Scheduling strategies, Priority structures and Task management, Real Time Clock Handler, Code sharing, Resource Control, Inter task Communication and Control , Example of Creating and RTOS based on Modula-2 kernel	30 Hours	1
III	Introduction to Design of Real Time Systems: Specification, Preliminary Design, Multitasking Approach, Monitors and Rendezvous, Fault Tolerance Techniques: Introduction, Faults, Fault types, Detection and Containment, Redundancy, Errors and Failures, Integrated Failure Handling.	30 Hours	1
IV	Introduction to Semaphores, Semaphore States, Types of Semaphores, Semaphores Implementation, Applications of RTOS, Semaphores in RTOS Applications, Examples: Creating Semaphores, Deleting Semaphores, Acquiring Semaphores, Releasing Semaphores	30 Hours	1

Text/Reference Books:

1. Silberschatz, Galvin and Gagne, "Operating Systems Concepts", Wiley
2. Real-Time Systems Design and Analysis, Phillip. A. Laplante, second edition, PHI, 2005.
3. Real Time Systems, Jane. W. S. Liu, Pearson education, 2005

BCS2033 Robotics

Course Objective:

1. This course provides an introduction to the mechanics of robots and spatial mechanics and motion planning.
2. The theoretical focus is on kinematics and dynamics of robotic manipulators and control design for non-linear mechanical systems.

3. Laboratory practice to learn simple robot programming.
4. This course will also expose students to some of the contemporary happenings in robotics, including current robotics research, applications, robot contests and robot web surfing.

Learning Outcome:

At the end of the course, the student should be able to:

1. Be familiar with the history, concept development and key components of robotics technologies.
2. Understand basic mathematic manipulations of spatial coordinate representation and transformation.
3. Understand and able to solve basic robot forward and inverse kinematics problems.
4. Understand and able to solve basic robotic dynamics, path planning and control problems.
5. Able to undertake practical robotics experiments that demonstrate the above skills.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction** Definition, Classification of Robots, geometric classification and control classification. Robot Elements: Drive system, control system, sensors, end effectors, gripper actuators and gripper design.	30 Hours	1
II	**Robot Coordinate Systems and Manipulator Kinematics** Robot Coordinate Systems and Manipulator Kinematics: Robot co-ordinate system representation, transformation, homogenous transform and its inverse, relating the robot to its world. Manipulators Kinematics, parameters of links and joints, kinematic chains, dynamics of kinematic chains, trajectory planning and control, advanced techniques of kinematics and dynamics of mechanical systems, parallel actuated and closed loop manipulators.	30 Hours	1
III	**Robot Control** Robot Control: Fundamental principles, classification, position, path velocity and force control systems, computed torque control, adaptive control, Seroo system for robot control, and introduction to robot vision, Robot Programming: Level of robot programming, language based programming, task level programming, robot programming synthesis, robot programming for welding, machine tools, material handling, assembly operations, collision free motion planning.	30 Hours	1
IV	**Applications** Applications: Application of robot in welding,	30 Hours	1

	machine tools, material handling, and assembly operations parts sorting and parts inspection.		

Text/Reference Books:

1. Coifet Chirroza, "An Introduction to Robot Technology" Kogan Page.
2. Y. Koren "Robotics for Engineers" Mcgraw Hill.
3. K. S. Fu, R.C. Gonzalez Y& CSG Lee, "Robotics" McGraw Hill.
4. J.J. Craig, "Robotics" Addison-Wesley.
5. Grover, Mitchell Weiss, Nagel Octrey, "Industrial Robots" Mcgraw Hill.
6. Asfahl, "Robots & Manufacturing Automation" Wily Eastern.

BCS2034 Computer Vision

Course Objective:

1. To introduce students with practice and theory of computer vision.
2. To give basics of pattern recognition concepts with application to computer vision.
3. Study about the concept of facial recognition system.
4. To provide students with necessary theory and skills for automatic analysis of digital images.
5. Learn basics of video processing and object recognition systems.

Learning Outcome:

At the end of the course, the student should be able to:

1. Understand the vision technology in conjunction with real world application.
2. Students will able to implement the functioning methods of surveillance cameras.
3. Apply all the enhancement methods of digital images.
4. Make automatic decision based on extracted feature information of images.
5. Understand the basic and commonly used paradigms of vision technology.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction to Recognition and Morphological Image Processing** Recognition Methodology; Conditioning, Labeling, Grouping, Extracting, Matching. Morphological Image Processing; Introduction, Dilation and Erosion, Opening and Closing, Hit-or-Miss transformation. Morphological algorithm operations on binary images; Morphological algorithm, Operations on gray-scale images, Thinning and Thickening, Region growing and region shrinking. Image Representation and Description. Representation schemes. Boundary descriptors. Region descriptors.	30 Hours	1
II	**Segmentation and Edge detection in Binary Machine Vision** Binary Machine Vision; Thresholding, Segmentation; Hierarchal segmentation, Rule-based segmentation,	30 Hours	1

	Motion-based segmentation. Connected component Labelling. Spatial Clustering; Split and Merge. Area Extraction in an image; Basic Concepts, Data Structures. Edge; Line Linking, Hough Transform. Line fitting and Curve fitting (Least-Square fitting).		
III	**Projection and Image Matching Techniques** Region Analysis; Region properties, External points, Spatial 05 moments, Mixed spatial gray-level moments. Boundary Analysis; Signature properties, Shape numbers. Facet Model Recognition; Labelling lines, Understanding line drawings. Classification of shapes by labelling of edges; Recognition of shapes, Consisting labelling problem, Back-tracking Algorithm. Projection; Perspective Projective geometry, Inverse perspective Projection, Photogrammetry - from 2D to 3D. Image matching; Intensity matching of ID signals, Matching of 2D image, Hierarchical image matching.	30 Hours	1
IV	**Object Modelling and Object Recognition** Object Models And Matching; 2D representation, Global vs. Local features. General Frame Works For Matching; Distance relational approach, Ordered structural matching, View class matching, Models database organization. Knowledge Based Vision; Knowledge representation, Control strategies, Information Integration. Object recognition; Hough transforms and other simple object recognition methods. Shape correspondence and shape matching. Principal component analysis(PCA). Shape priors for recognition	30 Hours	1

Text/Reference Books:

1. Robert Haralick and Linda Shapiro, "Computer and Robot Vision", Vol I, II, Addison- Wesley, 1993.
2. David A. Forsyth, Jean Ponce, "Computer Vision: A Modern Approach"
3. Milan Sonka,Vaclav Hlavac, Roger Boyle, "Image Processing, Analysis, and Machine Vision" Thomson Learning

Open Elective-I

OCS2002 Electronic Commerce

Course Objective:

1. The fundamentals of the business and economic motivations for e-Commerce as well as the needs and desires of individuals.

2. The underlying computation, information and communication environments that encompass and enable e-Commerce transactions, and
3. The evolving role of new highly portable, place-aware, always-with-you personal devices in e-Commerce, i.e., M-commerce

Learning Outcome:

At the end of the course, the student should be able to:

1. Understand the nature and trends in e-Commerce and Mobile commerce
2. Recognize the business impact and potential of e-Commerce
3. Explain the technologies required to make e-Commerce viable
4. Discuss the current drivers and inhibitors facing the business world in adopting and using e-Commerce
5. Explain the economic consequences of e-Commerce

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Politics, Science and Business of Sustainability:** Introduction to Electronic Commerce, What is E-commerce?, Traditional Commerce vs. E-commerce, Advantages and Disadvantages of E-commerce, Impact of E-commerce, Classification of E-commerce, Applications of E-commerce, Limitations of E-commerce, Electronic Commerce, Business Models, Native Content based Model, Transplated Content Model, Native Transaction Model, Transplated Transaction Models, Architectural Framework of Electronic Commerce, Network, Infrastructure, Information Distribution Technology, Networked Multimedia Content Publishing Technology, Security issues, Payment Services, Business Service Infrastructure, Public Policy and Legal Infrastructure	30 Hours	1
II	**Electronic Commerce Infrastructure:** Electronic Commerce: Network Infrastructure, Local Area Networks, Wide Area Networks, Internet, Domain Name Systems, Electronic Commerce: Information Distribution and Messaging, File Transfer Protocol (FTP) Application, Electronic Mail, World Wide Web Server, HTTP, Electronic Commerce: Information Publishing Technology, Information Publishing, Web Browser, Hypertext Markup Language, Common Gateway Interface, Multimedia Content	30 Hours	1
III	**Electronic Commerce Security and Payment System:** Securing the Business on Internet, Vulnerability of Information on Internet, Security Policy, Procedures and Practices, Site Security, Protecting the Network, Firewalls, Securing Network Transaction, Transaction Security, Cryptology,	30 Hours	1

	Cryptographic Algorithms, Public Key Algorithms, Authentication Protocols, Digital Signatures, Electronic Mail Security, Security Protocols for Web Commerce, Electronic Payment System, Introduction to Payment Systems, Online Payment Systems, Pre-Paid Electronic Payment Systems, Post-paid Electronic Payment Systems		
IV	**Mobile Commerce: Introduction, Framework, and Models:** What is Mobile Commerce?, Benefits of Mobile Commerce, Impediments in Mobile Commerce, Mobile Commerce Framework, Wireless Network Infrastructure, Information Distribution Protocols, Mobile Commerce Payment Systems, Mobile Payment Models, Mobile Commerce Applications	30 Hours	1

Text/Reference Books:

1. Bharat Bhaskar, "Electronic Commerce: Framework, Technologies & Applications 4/e", TMH
2. Ravi Kalakota, Andrew Winston, "Frontiers of Electronic Commerce", Addison-Wesley
3. Bajaj and Nag, "E-Commerce: The Cutting Edge of Business", Tata McGraw Hill
4. P. Loshin, John Vacca, "Electronic Commerce", Firewall Media, New Delhi
5. P. T. Joseph, "E-Commerce: An Indian Perspective", PHI Learning Pvt. Ltd.
6. Norman, Sadeh , "M Commerce: Technologies, Services, and Business Models", Wiley Computer Publishing

BCS2751 Web Technology Lab

Tools/Software Used: Netbeans / Sun Java Studio / JDK 1.6 or higher/visual studio 2010.

1. Create and validate of and sample XML document using DOM, SAX, and XSTL.
2. Create web pages using HTML, XHTML, Cascading Style sheets.
3. Using Java Script in different web pages for client side validation.
4. Develop sample application using Java Beans.
5. To create the several tags using HTML and display to the web browser.to design a Job registration form using XHTML.
6. To create a web page that displays college information using various style sheets using java or .net
7. To design the scientific calculator and make event for each button using java script.
8. 8. Develop sample application using javax. servelet package; handling Http Request & Responses, Using Cookies; - Session Tracking.
9. Create a sample application which accessing a Database from a webpage using java or .net.
10. Deploying JAVA Beans in a JSP Page

BCS2752 Network Security Lab

Tools/Software used: C/C++/Java

1. Write program for Mono alphabetic cipher.
2. Implementation of Play Fair cipher.
3. Implementation of Vigenere cipher (Polyalphabetic substitution).
4. Implementation of Hill cipher.
5. Implementation of Gauss cipher.
6. Implementation of Rail Fence cipher.
7. Implementation of S-DES algorithm for data encryption.
8. Implement RSA asymmetric (public key and private key)-Encryption. Encryption key (e, n) & (d,n)
9. Generate digital signature using Hash code.
10. Generate digital signature using MAC code.

BCS2801 Distributed Systems

Course Objective:

1. To impart basic knowledge of the issues concerning distributed systems, from both software and hardware viewpoints.
2. Have knowledge and understanding of the main principles, techniques and methods involved when dealing with distributed systems.
3. To get the knowledge of how distributed objects communicate by means of remote invocation.
4. To provide an in-depth overview of research topics in distributed systems

Learning Outcome:

At the end of the course, the student should be able to:

1. Understand the contrasting features between the distributed views of computing with the centralized one.
2. Understand in detail how distributed applications work and what requirements they aim to satisfy.
3. Understand in detail how distributed applications work and what issues and challenges they must contend with.
4. Understand in detail how distributed applications work and what architecture they exhibit.
5. Understand in detail how distributed applications work and what techniques and infrastructures they are built upon.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	Introduction to distributed systems: Definitions and Examples of Distributed systems; System Models: Architectural modelsand Fundamental models; limitations of distributed systems. Logical Clocks: Lamport's clocks, Vector logical clock, NTP; Message Passing System: Causal ordering of messages, Sates of a Distributed system, Local and Global State, Consistent and inconsistent states; Termination detection.	30 Hours	1
II	Mutual Exclusion: Requirements of Mutual	30	1

	Exclusion, Classification of distributed mutual exclusion: Non-token based, Quorum Based and Token Based mutual exclusion with examples; Performance metric for distributed mutual exclusion algorithms. Deadlock Detection: System models, Preliminaries, Deadlock prevention, Deadlock avoidance, Deadlock detection & resolution Agreement Protocols: Classification of Agreement Problem: Byzantine agreement problem, Consensus problem, Interactive consistency Problem; Solution to Byzantine Agreement problem; Application of Agreement problem.	Hours	
III	Resource Management : Distributed File Systems, Issues in distributed File System, Mechanism for building distributed file systems ; Distributed Shared Memory, Design issues in Distributed Shared Memory, Algorithm for Implementation of Distributed Shared Memory Failure Recovery: Backward and Forward recovery, Recovery in Concurrent systems: Checkpoints; Recovery in Distributed Database Systems; Fault Tolerance: Issues in Fault Tolerance, Voting Protocols. Transaction Control: Nested Transactions, Locks; Concurrency Control: Methods and their comparison, Concurrency control in Distributed Transactions, Concurrency control in distributed transactions; Replication: Fault tolerant services, Transactions with replicated data.	30 Hours	1

Text/Reference Books:

1. Singhal & Shivaratri, "Advanced Concept in Operating Systems", McGraw Hill
2. Ramakrishna, Gehrke," Database Management Systems", McGrawhill
3. Coulouris, Dollimore, Kindberg, "Distributed System: Concepts and Design", Pearson Education
4. Tenanuanbaum, Steen," Distributed Systems", Prentice Hall of India
5. Gerald Tel, "Distributed Algorithms", Cambridge University Press

Generic Elective – IV

BCS2041 Datamining and Warehousing

Course Objective:

1. Understand data mining principles and techniques.
2. Building basic terminology.
3. Learning how to gather and analyze large sets of data to gain useful business understanding.

4. Learning how to produce a quantitative analysis report/memo with the necessary information to make decisions.
5. Describing and demonstrating basic data mining algorithms, methods, and tools.
6. Identifying business applications of data mining.
7. Overview of the developing areas - web mining, text mining, and ethical aspects of data mining.
8. Develop and apply critical thinking, problem-solving, and decision-making skills.

Learning Outcome:

At the end of the course, the student should be able to:

1. Learn the concepts of database technology evolutionary path which has led to the need for data mining and its applications.
2. Examine the types of the data to be mined and present a general classification of tasks and primitives to integrate a data mining system.
3. Apply preprocessing statistical methods for any given raw data.
4. Explore DWH and OLAP, and devise efficient & cost effective methods for maintaining DWHs.
5. Discover interesting patterns from large amounts of data to analyze and extract patterns to solve problems, make predictions of outcomes.
6. Comprehend the roles that data mining plays in various fields and manipulate different data mining techniques.
7. Select and apply proper data mining algorithms to build analytical applications.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Data Mining** Overview , Motivation(for Data Mining),Data Mining, Definition & Functionalities, Data Processing, Form of Data Preprocessing ,Data Cleaning: Missing Values ,Noisy Data, Inconsistent Data, Data Integration and Transformation. , Data Reduction, Data Cube Aggregation, Dimensionality reduction, Data Compression, Numerosity Reduction, Clustering, Discretization and Concept Hierarchy generation.	30 Hours	1

II	**Data Mining Statistics and Association rule** Concept Description, Definition, Data Generalization, Analytical Characterization, Analysis of attribute relevance, Mining Class comparisons ,Statistical measures in large Databases, Measuring Central Tendency, Measuring Dispersion of Data ,Range ,Quartiles, Outliers, Box plots, Variance, Standard Deviation, Graph Displays of Basic Statistical class Description, Mining Association Rules in Large Databases, Association rule mining, Mining Single-Dimensional Boolean Association rules from transactional Databases– Apriori Algorithm, Mining Multilevel Association rules from Transaction Databases, Mining Multi-Dimensional Association rules from Relational Databases.	30 Hours	1
III	**Classification and Predictions** What is Classification & Prediction ,Issues regarding Classification and prediction, Decision tree, Bayesian Classification ,Classification by Back propagation, Multilayer feed-forward Neural Network, Back propagation Algorithm, Classification methods ,Knearest neighbor classifiers, Genetic Algorithm, Cluster Analysis ,Data types in cluster analysis, Categories of clustering methods, Partitioning methods, Hierarchical Clustering-,CURE and Chameleon, Density Based Methods-DBSCAN, OPTICS, Grid Based Methods-STING, CLIQUE ,Model Based Method –Statistical Approach, Neural Network approach, Outlier Analysis.	30 Hours	1
IV	**Data Warehousing and OLAP** Overview ,Definition, Delivery Process ,Difference between Database System and Data Warehouse, Multi-Dimensional Data Model, Data Cubes ,Stars ,Snow Flakes ,Fact Constellations ,Concept hierarchy, ProcessArchitecture,ThreeTierArchitecture,DataMarting, Aggregation,Historical information ,Query Facility, OLAP function and Tools, OLAP Servers, ROLAP, MOLAP, HOLAP, Data Mining interface, Security, Backup and Recovery, Tuning Data Warehouse, Testing Data Warehouse.	30 Hours	1

Text/Reference Books:

1. M.H. Dunham,"Data Mining: Introductory and Advanced Topics" Pearson Education
2. Jiawei Han, Micheline Kamber, "Data Mining Concepts & Techniques", Elsevier
3. Sam Anahory, Dennis Murray, "Data Warehousing in the Real World : A Practical Guide for Building Decision Support Systems, 1/e " Pearson Education
4. Mallach, "Data Warehousing System", McGraw –Hill

BCS2042 Artificial Neural Networks (ANN) and Fuzzy Logic

Course Objective:

1. To introduce the neural networks as means for computational learning.
2. To present the basic network architectures for classification and regression.
3. To provide knowledge for network tunning and over fitting avoidance.
4. Learn various methods of fuzzy computing and the basic concepts of neuro fuzzy systems.
5. Study the various methods of genetic algorithm and implement them into fuzzy environment.

Learning Outcome:

At the end of the course, the student should be able to:

1. Design single and multi-layer feed-forward neural networks and Back propagation networks.
2. Explain the differences between networks for supervised and unsupervised learning.
3. Comprehend neuro fuzzy modelling and demonstrate some applications of computational intelligence.
4. Understand the fuzzy approximate reasoning and fuzzy rule based systems.
5. Implement the genetic algorithms concept in heuristic search methods.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Neural Networks Introduction and Architecture** Structure of Biological neuron; Neuron, Nerve Structure and synapse. Artificial Neuron and its model. Activation functions. Neural network architecture; Single layer feed forward networks, multilayer feed forward networks, Recurrent networks, Technologies. Various learning techniques; Supervised learning, Un-Supervised learning, Reinforcement learning. Perception and convergence rule. Associative memory; Auto-associative memory, Hetro-associative memory.	30 Hours	1
II	**Neural networks - (BPN)** Back propagation network (BPN) Architecture. Perceptron model; Single layer artificial neural network, multilayer perception model, Linear Seperability. Back propagation learning methods; Effect of learning rule co-efficient, Back propagation algorithm; factors affecting back propagation training, Momentum Factor of EBPA. Application of Neural networks.	30 Hours	1
III	**Fuzzy logic** 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	30 Hours	1

	fuzzy logic. Fuzzy if-then rules. Fuzzy implications and Fuzzy algorithms. Fuzzyfications & Defuzzifications. Fuzzy Logic Controller (FLC).Industrial applications of Fuzzy.		
IV	**Genetic Algorithm(GA)** Introduction of Genetic Algorithm; Fundamentals of GA, Basic concepts of GA, Working principle, Encoding, fitness function, Reproduction. Genetic modelling; Inheritance operator, Cross over Operators, Inversion& deletion, Mutation operator, Bitwise operators. Generational Cycle of GA. Convergence of GA. Applications & advances in GA. Differences & similarities between GA & other traditional methods.	30 Hours	1

Text/Reference Books:

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 Netowrks"Prentice Hall of India
4. Timothy J. Ross, "Fuzzy Logic with Engineering Applications" Wiley I

BCS2043 Mobile Computing

Course Objective:

1. To provide guidelines, design principles and experience in developing applications for small, mobile devices, including an appreciation of context and location aware services
2. To develop an appreciation of interaction modalities with small, mobile devices (including interface design for non-standard display surfaces) through the implementation of simple applications and use cases.
3. To introduce wireless communication and networking principles, that support connectivity to cellular networks, wireless internet and sensor devices.
4. To understand the use of transaction and e-commerce principles over such devices to support mobile business concepts
5. To appreciate the social and ethical issues of mobile computing, including privacy.

Learning Outcome:

At the end of the module, the student will be able to:

1. Demonstrate a working understanding of the characteristics and limitations of mobile hardware devices including their user-interface modalities
2. Demonstrate the ability to develop applications that are mobile-device specific and demonstrate current practice in mobile computing contexts.
3. Demonstrate a comprehension and appreciation of the design and development of context-aware solutions for mobile devices.
4. Demonstrate an awareness of Generic and ethical issues, in particular those relating to security and privacy of user data and user behavior.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	Evolution from 2G over 3G to 4G: Beyond 3G Network Architectures, Overview: UMTS, HSPA (HSDPA and HSUPA) Introduction to Network Architecture, Air Interface and Radio Network, LTE-Introduction, Network Architecture, Air Interface and Radio Network, Basic Procedures, Summary and Comparison with HSPA.	30 Hours	1
II	802.16 WiMAX: Introduction, Network Architecture, The 802.16d Air Interface and Radio Network, The 802.16e Air Interface and Radio Network, Basic Procedures, Summary and Comparison with HSPA and LTE, 802.16m: Complying with IMT Advanced, 802.16j: Mobile Multi-hop Relay , 802.11Wi-Fi: Introduction, Network Architecture The Air Interface –from 802.11b to 802.11n Air Interface and Resource Management, Basic Procedures, Wi-Fi Security, Quality of Service: 802.11e.	30 Hours	1
III	Network Capacity and Usage Scenarios:Usage in Developed Markets and Emerging Economies, How to Control Mobile Usage, Per Minute Charging, Volume Charging, Split Charging, Small-screen Flat Rates Strategies to Inform Users When Their Subscribed Data, Measuring Mobile Usage from a Financial Point of View, Cell Capacity in Downlink, Current and Future Frequency Bands for Cellular Wireless Cell Capacity in Uplink, Per-user Throughput in Downlink, Per-user Throughput in the Uplink Traffic Estimation Per user, Overall Wireless Network Capacity, Network Capacity for Train Routes, Highways and Remote Areas, A Hybrid Cellular/Wi-Fi Network for the Future	30 Hours	1
IV	Voice over Wireless, Circuit-switched Mobile Voice Telephony: Circuit Switching, A Voice-optimized Radio Network, The Pros of Circuit Switching, Packet-switched Voice Telephony, Network and Applications are Separate in Packet-Switched Networks, Wireless Network Architecture for Transporting IP packets, Benefits of Migrating Voice Telephony to IP,Voice Telephony Evolution and Service Integration, Voice Telephony over IP: the End of the Operator Monopoly, SIP Telephony over IP network.	30 Hours	1

Text/Reference Books:

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

BCS2044 Natural Language Processing

Course Objective:

1. Develop familiarity with lexical, syntactic, semantic and pragmatic aspects of NLP
2. Understand the basic concepts of context free grammar and Parsing methods of NLP.
3. Study the basic problem regarding NLP and removal methods of it.
4. Develop an understanding of NLP Models and Algorithms.
5. Develop background in statistical and machine learning approaches to NLP.

Learning Outcome:

At the end of the course, the student should be able to:

1. Show sensitivity to linguistic phenomena and an ability to model them with formal grammars.
2. Understand and carry out proper experimental methodology for training and evaluating empirical NLP systems.
3. Be able to manipulate probabilities, construct statistical models over strings and trees, and estimate parameters using supervised and unsupervised training methods.
4. Apply NLP algorithms into Machine Learning Techniques.
5. Be able to design, implement, and analyse NLP algorithms.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction of NLP** Introduction to Natural Language Understanding; The study of Language, Applications of NLP, Evaluating Language Understanding Systems. Different levels of Language Analysis; Representations and Understanding. Organization of Natural language Understanding Systems. Linguistic Background. An outline of English syntax.	30 Hours	1
II	**Knowledge Representation schemes in NLP** Introduction to semantics and knowledge representation, Some applications, machine translation, database interface. Grammars and Parsing, Grammars and sentence Structure, Top-Down and Bottom-Up Parsers, Top-Down Chart Parsing. Transition Network Grammars. Feature Systems and Augmented Grammars; Basic Feature system for English, Morphological Analysis	30 Hours	1

	and the Lexicon, Parsing with Features. Augmented Transition Networks (ATN).		
III	**Grammar and Parsing Techniques in NLP** Grammars for Natural Language, Auxiliary Verbs and Verb Phrases, Movement Phenomenon in Language. Handling questions in Context-Free Grammars. Human preferences in Parsing. Encoding uncertainty. Deterministic Parser.	30 Hours	1
IV	**Ambiguity Resolution and Probabilistic grammar** Ambiguity Resolution in NLP, Statistical Methods, Probabilistic Language Processing, Estimating Probabilities. Part-of- Speech tagging. Obtaining Lexical Probabilities. Probabilistic Context-Free Grammars. Best First Parsing. Semantics and Logical Form. Word senses and Ambiguity. Encoding Ambiguity in Logical Form.	30 Hours	1

Text/Reference Books:

1. Akshar Bharti, Vineet Chaitanya and Rajeev Sangal, "NLP: A Paninian Perspective", Prentice Hall,New Delhi.
2. James Allen, "Natural Language Understanding", 2/e, Pearson Education, 2003.
3. D. Jurafsky, J. H. Martin, "Speech and Language Processing", Pearson Education, 2002.
4. L.M. Ivansca, S. C. Shapiro, "Natural Language Processing and Language Representation".

Generic Elective – V

BCS2051 Data Compression

Course Objective:

1. To provide students with contemporary knowledge in Data Compression and Coding.
2. To equip students with skills to analyze and evaluate different Data Compression and Coding methods.
3. Student knows basic algorithms used in lossless and lossy compression.
4. Student knows basic mathematical models used in lossless and lossy compression.

Learning Outcome:

At the end of the course, the student should be able to:

1. Display competence in the fundamental ideas of the lossless data compression.
2. Demonstrate understanding of the underlying mathematical theory and algorithms.
3. Understand fundamental ideas of quantization and transform coding.
4. Understand how lossless and lossy compression algorithms can be used for solving scientific and engineering problems.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction** Compression Techniques: Lossless and Lossy Compression, Measures of performance, Modelling and Coding; Mathematical Preliminaries for Lossless compression: A brief introduction to information theory; Models: Physical models, Probability models, Markov models, Composite source model ;Coding :Uniquely decodable codes, Prefix codes.	30 Hours	1
II	**Huffman and Arithmetic Coding** Huffman coding algorithm: Minimum variance Huffman codes; Adaptive Huffman coding: Update procedure, Encoding procedure, Decoding procedure; Golomb codes, Rice codes, Tunstall codes Applications of Huffman coding: Lossless image compression, Text compression and Audio Compression Arithmetic Coding: Introduction, Coding a Sequence, Generating a Tag, Deciphering a Tag, Comparison of Huffman and Arithmetic Coding, Applications.	30 Hours	1
III	**Dictionary Coding and Context Based Compression** Dictionary Techniques: Introduction, Static Dictionary: Diagram Coding; Adaptive Dictionary: The LZ77Approach ,TheLZ78Approach, Applications: File Compression-UNIX compress, Image Compression-Graphics Interchange Format (GIF) and Portable Network Graphics (PNG),Compression over modems-V.42 bits . Context Based Compression: Introduction , Prediction with Partial Match (ppm)-The basic algorithm, The ESCAPE SYMBOL, Length of context, The Exclusion Principle; The Burrows-Wheeler Transform: Move-to-front coding ,Dynamic Markov Compression. Lossless image compression: Introduction, CALIC, JPEG-LS, Multi-resolution Approaches, Facsimile Encoding.	30 Hours	1
IV	**Mathematical Preliminaries for Lossy Coding, Scalar and Vector Quantization** Mathematical Preliminaries for Lossy Coding: Introduction, Distortion criteria, Models. Scalar Quantization: Introduction, The Quantization Problem, Uniform Quantizer, Adaptive Quantization, Non- uniform Quantization Vector Quantization: Introduction, Advantages of Vector Quantization *over* Scalar Quantization, The	30 Hours	1

	Linde -Buzo- Gray Algorithm, Tree structured Vector Quantizers.		

Text/Reference Books:

1. Data Compression – David Salomon , Springer Publication, 4th Edition.
2. Introduction to Data Compression – Khalid Sayood, Morgan Kaufmann Series, 3rd Edition

BCS2052 Digital Image Processing

Course Objective:

1. Cover the basic theory and algorithms that are widely used in digital image processing
2. Expose students to current technologies and issues that are specific to image processing systems
3. Develop hands-on experience in using computers to process images

Learning Outcome:

1. Understand image formation and the role human visual system plays in perception of gray and color image data.
2. Get broad exposure to and understanding of various applications of image processing in industry, medicine, and defense.
3. Learn the signal processing algorithms and techniques in image enhancement and image restoration.
4. Acquire an appreciation for the image processing issues and techniques and be able to apply these techniques to real world problems.
5. Be able to conduct independent study and analysis of image processing problems and techniques.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction & Fundamentals** Introduction: Motivation and Perspective, Applications, Components of Image Processing System. Fundamentals: Element of Visual Perception, A Simple Image Model, Sampling and Quantization; Image Enhancement in Spatial Domain Introduction; Basic Gray Level Functions: Piecewise-Linear Transformation Functions-Contrast Stretching; Histogram Specification: Histogram Equalization, Local Enhancement, Enhancement using Arithmetic/Logic Operations-Image Subtraction, Image Averaging; Basics of Spatial Filtering: Smoothing - Mean filter, Ordered Statistic Filter; Sharpening – The Laplacian;	30 Hours	1
II	**Image Enhancement in Frequency Domain Fourier Transform and the Frequency Domain & Image Restoration** Basis of Filtering in Frequency Domain: Filters, Low-	30 Hours	1

	pass, High-pass, Correspondence Between Filtering in Spatial and Frequency Domain, Smoothing Frequency Domain Filters-Gaussian Lowpass Filters; Sharpening Frequency Domain Filters-Gaussian Highpass Filters; Homomorphic Filtering. Image Restoration: A Model of Restoration Process, Noise Models, Restoration in the presence of Noise only-Spatial Filtering-Mean Filters: Arithmetic Mean filter, Geometric Mean Filter, Order Statistic Filters – Median Filter, Max and Min filters; Periodic Noise Reduction by Frequency Domain Filtering- Bandpass Filters; Minimum Mean-square Error Restoration.		
III	**Colour& Morphological Image Processing &Registration** Color Fundamentals: Color Models- Converting Colors to different models; Color Transformation, Smoothing and Sharpening, Color Segmentation. Morphological Image Processing: Introduction, Logic Operations involving Binary Images, Dilation and Erosion, Opening and Closing, Morphological Algorithms- Boundary Extraction, Region Filling, Extraction of Connected Components, Convex Hull, Thinning, Thickening. Registration: Introduction, Geometric Transformation-Plane to Plane transformation; Mapping, Stereo Imaging-Algorithms to Establish Correspondence; Algorithms to Recover Depth.	30 Hours	1
IV	**Segmentation & Object Recognition** Segmentation: Introduction, Region Extraction, Pixel-Based Approach, Multi-level Thresholding, Local Thresholding, Region-based Approach, Edge and Line Detection-Edge Detection, Edge Operators, Pattern Fitting Approach, Edge Linking and Edge Following, Edge Elements. Feature Extraction Representation: Topological Attributes, Geometric Attributes. Description: Boundary-based Description, Region-based Description, Relationship. Object Recognition: Deterministic Methods, Clustering, Statistical Classification, Syntactic Recognition, Tree Search, Graph Matching.	30 Hours	1

Text/ Reference Books:

1. Digital Image Processing 2nd Edition, Rafael C. Gonzalvez and Richard E. Woods. Published by: Pearson Education.
2. Digital Image Processing and Computer Vision, R.J. Schalkoff. Published by: John Wiley and Sons, NY.
3. Fundamentals of Digital Image Processing, A.K. Jain. Published by Prentice Hall, Upper Saddle River, NJ.

BCS2053 Bioinformatics

Course Objective:

1. Study to develop methods and software tools for understanding biological data.
2. To introduce students to the fundamentals of evolution, molecular biology, and molecular evolution.
3. To study DNA, RNA important molecules, protein data, etc. their structure, replication and transcription.
4. Study on biological databases which help in analyzing biological data and their interpretation.
5. Introduction of Perl programming for developing tools for genome sequencing and other applications.

Learning Outcome:

After completing the course, the students should be able to:

1. Have a good working knowledge of basic bioinformatics tools and databases such as GenBank, BLAST, multiple alignment, and phylogenetic tree construction.
2. Describe the contents and propeties of the most important bioinformatical databases, perform text- and sequence-based searches, and analyse and discuss the results in light of molecular biological knowledge.
3. Explain the major steps in pairwise and multiple sequence alignment, explains the principle for, and executes pairwise sequence alignment by dynamic programming.
4. give examples of methods for describing and analysing genes, genomes and gene expression
5. explain the major features of methods for modelling protein structures and use programs for visualizing and analysing such structures

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Introduction** Bioinformatics objectives and overviews, Interdisciplinary nature of Bioinformatics, Data integration, Data analysis, Major Bioinformatics databases and tools. Metadata: Summary & reference systems, finding new type of data online. Molecular Biology and Bioinformatics: Systems approach in biology, Central dogma of molecular biology, problems in molecular approach and the bioinformatics approach, Overview of the bioinformatics applications.	30 Hours	1
II	**The Information Molecules and Information Flow** Basic chemistry of nucleic acids, Structure of DNA, Structure of RNA, DNA Replication, -Transcription, -Translation, Genes- the functional elements in DNA, Analyzing DNA,DNA sequencing. Proteins: Amino acids, Protein structure, Secondary, Tertiary and	30 Hours	1

	Quaternary structure, Protein folding and function, Nucleic acid-Protein interaction; Perl: Perl Basics, Perl applications for bioinformatics- Bioperl, Linux Operating System, Understanding and Using Biological Databases, Java clients, CORBA, Introduction to biostatics		
III	**Nucleotide sequence data** Genome, Genomic sequencing, expressed sequence tags, gene expression, transcription factor binding sites and single nucleotide polymorphism. Computational representations of molecular biological data storage techniques: databases (flat, relational and object oriented), and controlled vocabularies, general data retrieval techniques: indices, Boolean search, fuzzy search and neighboring, application to biological data warehouses.	30 Hours	1
IV	**Biological data types and their special requirements** Sequences, macromolecular structures, chemical compounds, generic variability and its connection to clinical data. Representation of patterns and relationships: alignments, regular expressions, hierarchies and graphical models.	30 Hours	1

Text/Reference Books:

1. O'Reilly, " Developing Bio informatics computer skills", Indian Edition's publication
2. Rastogi, Mendiratta, Rastogi, "Bioinformatics concepts, skills & Applications", CBS Publishers
3. Rashidi, Hooman and Lukas K. Buehler, "Bioinformatics Basic Applications" CRC Press.
4. "Bioinformatics" , Addison Wesley
5. Stephen Misner& Stephen Krawetz, " Bioinformatics- Methods & Protocols"
6. Cynthia Gibas and Per Jambeck, Introduction to Bioinformatics computer Skills, 2001 SPD
7. Atwood, Introduction to Bioinformatics, Person Education
8. James Tisdall, Beginning Perl for Bioinformatics, SPD

BCS2054 Pattern Recognition

Course Objective:

1. Learn the fundamental concepts and applications of pattern recognition.
2. Learn the concepts of Bayes decision theory.
3. Understand the concepts of linear and nonlinear classifiers.
4. Understand the concepts of feature selection and generation techniques.
5. Understand the concepts of supervised learning and system evaluation.
6. Understand the concepts of unsupervised learning and clustering algorithms.
7. Develop some applications of pattern recognition.

Learning Outcome:

At the end of the course, the student should be able to:

1. Understand the fundamental pattern recognition and machine learning theories.
2. Design and implement certain important pattern recognition techniques.
3. Applying the pattern recognition theories to applications of interest.
4. Design systems and algorithms for pattern recognition (signal classification), with focus on sequences of patterns.
5. Analyze classification problems probabilistically and estimate classifier performance.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	Introduction to Pattern Recognition,Basics of pattern recognition**,**Design principles of pattern recognition system**,**Learning and adaptation**,**Pattern recognition approaches**,**Mathematical foundations – Linear algebra ,Probability Theory**,**Expectation, mean and covariance**,**Normal distribution**,**Multivariate normal densities**,**Chi squared test.	30 Hours	1
II	Statistical Patten Recognition**,**Bayesian Decision Theory, Classifiers**,**Normal density **,**Discriminant functions.	30 Hours	1
III	Parameter Estimation Methods**,**Maximum-Likelihood estimation**,**Bayesian Parameter estimation**,**Dimension reduction methods**,**Principal Component Analysis (PCA)**,**Fisher Linear discriminate analysis**,**Expectation-maximization (EM)**,**Hidden Markov Models (HMM)**,**Gaussian mixture models	30 Hours	1
IV	Nonparametric Techniques and Unsupervised Learning,Density Estimation, Parzen Windows,K-Nearest Neighbor Estimation,Nearest Neighbor Rule,Fuzzy classification,Clustering,Criterion functions for clustering,Clustering Techniques,Iterative square - error partitioned clustering – K means, Agglomerative hierarchical clustering, Cluster validation.	30 Hours	1

Text/Reference Books:

1. Richard O. Duda, Peter E. Hart and David G. Stork, “Pattern Classification”, 2nd Edition, John Wiley, 2006.
2. C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2009.
3. S. Theodoridis and K. Koutroumbas, “Pattern Recognition”, 4th Edition, Academic Press, 2009.

Open Elective-II

OCS2001 Human Computer Interaction

Course Objective:

1. Learn basic concepts of designing of interface devices.
2. Recognize how a computer system may be modified to include human diversity.
3. Select an effective design style for a specific application.
4. Demonstrate an understanding of guidelines, principles, and theories influencing human computer interaction.
5. Learn the different levels of interfacing devices for the purpose of human easiness.

Learning Outcome:

At the end of the course, the student should be able to:

1. Design mock ups and carry out user and expert evaluation of interfaces.
2. Carry out the steps of experimental design, usability and experimental testing, and evaluation of
3. Human computer interaction systems.
4. Use the information sources available, and be aware of the methodologies and technologies
5. Supporting advances in HCI.
6. Able to build a basic prototype of interactive devices.

Course Contents:

Module	Course Topics	Total Hours	Credits
I	**Essentials and framework for designing interactive systems** Essentials of designing interactive systems. Designing interactive systems; A fusion of skills. PACT: A framework for designing interactive systems; Peoples, Activities, Contexts, Technologies. The process of human-centred interactive systems design. Usability. Experience design. The Home Information Centre (HIC); A case study in designing interactive systems.	30 Hours	1
II	**Techniques for designing interactive Systems** Techniques for designing interactive systems; Understanding, Envisionment, Design, Evaluation, Task analysis. Contextual Design; The contextual interview and work modelling; From models to design. Interface design; Visual Aspects, Multimodality, Mixed Reality.	30 Hours	1
III	**Interface Designing of interactive systems** Design process; Human interaction with computers, Importance of human characteristics, Human consideration. Human interaction speeds. Understanding business junctions. Components of	30 Hours	1

	interface; Text and messages. Icons and increases in interface; Multimedia, Colors, Uses problems, Choosing colors.		
IV	**Contexts for designing interactive systems and Introduction to Windows** Contexts for designing interactive systems; Designing websites, Web 2.0. CSCW; working in groups. Agents and avatars. Ubiquitous computing. Mobile computing. Screen Designing; Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content. Technological consideration in interface design. Foundations of designing interactive systems; Memory and attention; Emotion and affective computing, Cognition and action, Social interaction, Perception and navigation. Introduction of Windows; New and Navigation schemes selection of window, Selection of devices based and screen based controls.	30 Hours	1

Text/Reference Books:

1. David Benyon, "Designing Interactive Systems: A comprehensive guide to HCI and interaction
   design", (Second Edition), Pearson Education Limited, 2010.
2. Andrew Sears , Julie A. Jacko, "The Human-Computer Interaction Handbook: Fundamentals,
3. Evolving Technologies and Emerging Applications", Second Edition, CRC Press, 2007
4. "The essential guide to user interface design", Wilbert O Galitz, Wiley DreamaTech.
5. "Designing the user interface"; 3rd Edition Ben Shneidermann , Pearson Education Asia.

BCS2851 Distributed System Lab

Tools/Software Used:

a) OS: Linux/Windows
b) IDE: Netbeans, JavaTM 1.6.x or higher, preferably from Sun, must be installed.
c) ssh must be installed and sshd must be running to use the Hadoop scripts that manage remote Hadoop daemons.
d) Hadoop Cluster Setup; Aglets 2.5: open source platform Used to create agents in java.

Note: The minimum of 8 experiments are to be performed from the following.

1. Design a Distributed Application using RMI for remote computation; where client submits two strings to the server and server returns the concatenation of the given strings.
2. Design a Distributed Application using Message passing Interface for remote computation.

3. Design a Distributed application using socket. Application consist of a server which takes an integer value from the client calculates factorial and returns the result to the Client program.
4. Design a Distributed Application using Mobile agent that travels in the network and performs a given task on the targeted node. You may assign any task to the agent for example to carry out afile reading/processing at the remote machine and so on.
5. Develop a Distributed Application using MapReduce for finding the coolest year from the available Weather data.
6. Find out the list of users who owns a file having maximum size in the current working directory using Map Reduce Program.
7. Design a distributed application using RMI.
8. Design a distributed application which consists of a statefull server using socket primitives.
9. Design a distributed application which consists of a server and client using threads.
10. Write a MapReduce Application which processes weather data. List out the coldest years from the available data. Use sample weather data available from the internet or prepare it referring the input discussed in the lecture and process it using a pseduo distribution mode on hadoop platform.
11. Design a distributed application using RMI. Where client submits a string to the server and server returns the reverse of it.
12. Design a distributed application which consists of a stateless server using socket primitives.
13. Design a distributed application which consists of a server and client using threads.