M.Sc. in Computer Science (MCS)

Distribution of Credits

Core (PC)	Major Electives (PE)	Minor Electives (OE)	Total Credits
54	**27**	**9**	**90**

Semester-wise Distribution of Credits

Semester I	Semester II	Semester III	Semester IV	Total Credits (min)
22	**22**	**24**	**22**	**90**

Semester-wise Scheduling of Courses

Semester I

Course No.	Course Title	Type (PC/PE/OE)	L-T-P	Credits
MCS-111	***Theory of Computation***	PC	3-0-4	5
MCS-112	**Software Systems Lab**	PC	0-0-8	4
MCS-121	Computer Network	PE	3-0-2	4
MCS-122	Software Engineering	PE	3-1-4	6
MCS-131	*Elementary Statistical Theory and Method/ Object Oriented Programming/Web Technology*	OE	3-0-0	3
Total Credits	*4 Lecture Courses, 1 Lab Course*	*PC=9, PE=10, OE=3*	12-1-18	**22**

Semester II

Course No.	Course Title	Type (PC/PE/OE)	L-T-P	Credits
MCS-211	**Architecture of High Performance Computer Systems**	PC	3-0-2	4
MCS-212	**Compiler Design**	PC	3-0-2	4
MCS-221	Computer Graphics	PE	3-0-6	6
MCS-222	Operation Research	PE	3-0-4	5
MCS-231	*Bioinformatics/ Neural Computing*	OE	3-0-0	3

Total Credits	*5 Lecture courses*	*PC=8, PE=11, OE=3*	15-0-14	**22**

Semester III

Course No.	Course Title	Type (PC/PE/OE)	L-T-P	Credits
MCS-311	**Algorithms**	PC	3-0-6	6
MCS-312	**Artificial Intelligence**	PC	3-0-6	6
MCS-341	**Independent Study**	PC	0-3-0	3
MCS-321	Simulation & Modeling	PE	3-0-6	6
MCS-331	*Parallel Computing/Distributed Computing*	OE	3-0-0	3
Total Credits	*4 Lecture courses, 1Tutorial*	PC=15, PE=6, OE=3	12-3-18	**24**

Semester IV

Course No.	Course Title	Type (PC/PE/OE)	L-T-P	Credits
MCS-451	**Major Project & Dissertation**	PC	0-0-36	18
MCS-452	**Comprehensive Viva**	PC	-	4
Total Credits	*1 project, Viva*	*PC=22*	*0-0-36*	22

Course MCS-111 Theory of Computation

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course No**	MCS-111
3.	**L-T-P structure**	3-0-4
4.	**Credits**	5
5.	**Course Title**	Theory of Computation
6.	**Prerequisites**	Programming Languages
7.	**Status**	Core
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes

9. **Frequency of offering** ODD Semester ANNUAL

10. **Course objective**

To introduce the concept of computer theory

11. **Course contents**

A brief review of Finite Automata, Regular expressions, Regular languages, Deterministic and non-deterministic computations. Pumping lemma for regular languages, Context free languages, Pushdown automaton, Pumping lemma for Context free languages, Context Sensitive languages, and Grammar types. Turing machines (TM), Post machines, Variations of TM's, Universal Turing machines (UTM), Church's thesis, Relation of languages to automata. Introduction to recursive function theory; Turing computable functions, Halting problem, Solvability and undecidability. Computability and complexity theory.

12. **Brief description of laboratory activities**

Simulating programs for computing machines, testing of grammars etc.

13. **Suggested texts and reference materials**

i. J.E.Hopcraft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computation", Pearson Education.

ii. Cohen, "Introduction to Computer Theory", John Wiley.

14. **Resources required for the course** C and C++ compilers

Course MCS-112 Software Systems Laboratory

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course No**	MCS-112
3.	**L-T-P structure**	0-0-8
4.	**Credits**	4
5.	**Course Title**	Software systems laboratory
6.	**Prerequisites**	Nil
7.	**Status**	Core
8.	**Overlap with other UG/PG courses from other Departments/Centers**	No

9. **Frequency of offering** ODD semester ANNUAL

10 **Course objective**

The objective of this practical activity course is to equip the students with general software design and development skills and making them familiar with standard development and maintenance tools

11 **Course contents**

A set of project oriented assignments, which will be announced at the start of semester with definite submission deadlines. The set of assignments will be designed to develop skills and familiarity with a majority of the following: make configuration management tools, installation of software, archiving and creation of libraries, version control systems, documentation and literate programming systems (noweb and LaTeX), lex, yacc, perl and other scripting languages, sockets and RPCs, usage of standard libraries like pthreads, numerical packages, XML and semi-structured data, simulation environments, testing and validation tools depending upon the facilities and resources available in the department

12 **Suggested texts and reference materials**

Manuals and hand-outs

Course MCS-121 Computer Networks

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-121
3.	**L-T-P structure**	3-0-2
4.	**Credits**	4
5.	**Course Title**	Computer Networks - Theory and Applications
6.	**Prerequisites**	Computer Organization
7.	**Status**	Major Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes

9. **Frequency of offering** ODD semester ANNUAL

10. **Course objective**

To appreciate the historical evolution of Computer Networks to understand the potential of technologies currently available, to be creative in the deployment of existing technology conforming to standards. To be able to create new technologies to meet the challenges of emerging requirements

11. **Course contents**

Fundamentals of Digital communications, channel capacity, bit error rate, media characteristics, FDM, TDM, TDD, FDD, CDMA, Statistical Multiplexing; Framing and Synchronization; Point to point and broadcast communications, Multi access protocols: Aloha, CSMA and its variations, Token Ring; Error Control Techniques; Flow control; Bridges, Repeaters, Switches and the spanning tree protocol. Their concepts, capabilities, standards and performance. Network: Routing, Congestion control, Internet protocols; Multicast and mobile routing. Current trends in high-speed networking (ATM, Gigabit) - Current trends in high-speed transmission technology (Sonet, SDH, Cellular) - Learning models for characterization of sources

12. **Brief description of laboratory activities**

Network simulation and performance evaluation

13. **Suggested texts and reference materials**

i. Andrew S. Tanenbaum, Computer Networks, 3rd Edition, Prentice Hall of India, 1996.

ii. William Stallings, Hand Book of Data Communication, Volumes I, II and III, JW, 1990

iii. Ulyss Black, Computer Networks, Prentice Hall of India, 1987

iv. S.V. Raghavan and S.K. Tripathi, Networked Multimedia Systems: Concepts, Architecture and Design, Prentice Hall of India, 1998

v. Peterson, computer Networks, Kauffman and Moran press, 1997

vi. Keshav, An Engineering Approach to Computer Networks, Addison & Wesley, 1998

vii. Bertsakes and Gallagher, Data Networks, Prentice Hall of India, 1987.

14. **Resources required for the course** Network Simulator Software

Course MCS-122 Software Engineering

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-122
3.	**L-T-P structure**	3-1-4
4.	**Credits**	6
5.	**Course Title**	Software Engineering
6.	**Prerequisites**	Data Structures, Programming Languages
7.	**Status**	Major Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	ODD semester ANNUAL

10. **Course objective:** *To develop skills and conceptual framework for undertaking large software project and managing software and software projects*

11. **Course contents**

Introduction to Software Engineering: Definition, Software development and life-cycle models. Requirements specification and analysis. Top-down design and development. Information hiding, abstraction, modularity, object-oriented techniques. Separate compilation, configuration management, program libraries. Design patterns; UML. Documentation. Validation. Quality assurance, software reliability, safety. Testing and test case generation. Software metrics. Cost analysis and estimation, manpower and time management. Organization and management of large software design projects.

12. **Brief description of laboratory activities :** Project oriented activity involving extensive use of CASE tools

13. **Suggested texts and reference materials**

i. Ian Sommerville, "Software Engineering", Addison-Wesley, 1999

ii. Peters and Pedrycz, "Software Engineering: an Engineering Approach", Wiley,1999.

iii. R.S. Pressman, Software Engineering, McGraw Hill, 1996.

14.	**Resources required for the course**	Usual PC and software requirements. CASE tools.

15. **Lecture outline with topics and number of lectures**

Topics	Number of lectures
Introduction	4
Specification and analysis	8
Preliminary and detailed design methodology	6
UML, design patterns, CASE tools and techniques	6
Validation, testing, quality and safety assurance	6
Metrics, estimation, etc.	6
Documentation, installation, pragmatics	3
Organization	3

Course MCS-131 Object Oriented Programming

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-131
3.	**L-T-P structure**	3-0-0
4.	**Credits**	3
5.	**Course Title**	*Object Oriented Programming*
6.	**Prerequisites**	-
7.	**Status**	Minor Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	ODD semester ANNUAL

10. **Course objective**

11. **Course contents :** General Concepts, Introduction to Object-oriented programming paradigm and design. Object, class, superclass, subclass, metaclass, Hierarchy, instance, polymorphism (Operator Overloading), Inheritance: Hierarchical, Multiple, Selective. Object Oriented Methods: Object oriented analysis, Construction and Testing. Object Modeling techniques, case studies. Introduction to OOP Languages, C++, JAVA.Use of OOP concepts in different areas, In Software Engineering, In Operating Systems, In Object oriented Database, In object oriented graphics.

12. **Suggested texts and reference materials**

i. Booch, G., “Object Oriented Analysis and Design”, Benjamin/Cummins Publishing Co., Redwood City, USA, 1994.

ii. Rebecca et.al. “Designing Object oriented Software”, PHI, 1996.

iii. Rumbaugh, J. et.al., “Object Oriented Modelling and Design”

Course MCS-131 Web Technology

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-131
3.	**L-T-P structure**	3-0-0
4.	**Credits**	3
5.	**Course Title**	Web Technology
6.	**Prerequisites**	-
7.	**Status**	Minor Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	ODD semester ANNUAL

10. **Course objective**

11. **Course contents :** The Internet Client server software models, world wide web & web browsers, HTML building blocks, search engines. Advanced web page construction: Image file, JAVA scripts, applets, query and query refinements, software on internet, internet relay chat etc. Java language introduction, object references, instance variables, dot operator constructors, Methods overloading, Inheritance, Exception handling Threads and Synchronization, utilities, I/O, Networking in Java, server socket, URLK, URL connection, Abstract Window, JDK.

12. **Suggested texts and reference materials :**

i. Elizabeth Castro, “HTML for the World Wide Web”, Peachpit Press Pearson Education.

ii. Lehnert Wendy, “Web 101, Making the network for you”, Pearson Education, Asia.

iii. Naughton Patrick, “The JAVA Hanbook”, TataMcgraw Hill 1996.

iv. Winston PH & Narsimhan, “On to JAVA 1.2”, Addison Wesley.

Course MCS-131 Elementary Statistical theory and Methods

13.	**Department proposing the Course**	Department of Computer Science
14.	**Course Number**	MCS-131
15.	**L-T-P structure**	3-0-0
16.	**Credits**	3
17.	**Course Title**	Elementary Statistical theory and Methods
18.	**Prerequisites**	-
19.	**Status**	Minor Elective
20.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
21.	**Frequency of offering**	ODD semester ANNUAL

22. **Course objective**

As prerequisites for Operation Research and Simulation & Modeling

23. **Course contents :** Probability, Random Variable, Distribution of Random Variable: Expected Value and other Properties Tests of Randomness and Goodness of Fit, Comparison of Means and Variances (one, two and k samples), One way and two-way ANOVA, Tests of Correlation and Regression. Non-parametric Tests Elements of Stochastic Process: Markov Chain, Chapman–Kolmogorov Equation, Markov Process, Poisson Process, Birth Death Process Computer solution of the above problems

24. **Suggested texts and reference materials**

i. MR Spiegel :Probability and Statistics, Schaum Series

ii. PL Meyer :Intrductory Probability and Statistical Applications, Addison–Wesley Publishing Co. Pvt. Ltd

iii. W Feller: An Introduction to Probability Theory& Its Applications

iv. AM Goon, MK Gupta, B Dasgupta: An Outline of Statistical Theory Vol. 1, The World Press Private Ltd

v. PG Hoel : Introduction to Mathematical Statistics, John Wiley & Sons

vi. JA Payne Introduction to Simulation, Programming Techniques and Methods of Analysis, Tata McGraw Hill Publishing Co. Ltd, 1988

25. **Lecture outline with topics and number of lectures**

Topics	**Number of lectures**
Probability, Random Variable, Distribution of Random Variable	12
Tests of Randomness and Goodness etc.	10
Elements of Stochastic Process	8

Computer solution	5

Course MCS-211: Architecture of High Performance Computer Systems

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-211
3.	**L-T-P structure**	3-0-2
4.	**Credits**	4
5.	**Course Title**	Architecture of High Performance Computer Systems
6.	**Prerequisites**	Computer Organization
7.	**Status**	Core Course
8.	**Overlap with other UG/PG courses from other Departments/Centers**	No
9.	**Frequency of offering**	EVEN semester ANNUAL

10. **Course objective :** *This course is designed as a follow up of a basic course in Computer Architecture. The objective is to discuss the advanced architectural concepts, which improve the performance of computer systems. Instruction level as well as system level parallelism are considered*

11. **Course contents :** Classification of parallel computing structures; Instruction level parallelism - static and dynamic pipelining, improving branch performance, super-scalar and VLIW processors; High performance memory system; Shared memory multiprocessors and cache coherence; Multiprocessor interconnection networks; Performance modeling; Issues in programming multiprocessors; Data parallel architectures

12. **Brief description of laboratory activities :** Use of performance evaluation/simulation tools to compare architectures Implementing models of architectural features and studying their performance

13. **Suggested texts and reference materials**

    i. D. Sima, T. Fountain, P. Kacsuk, "Advanced Computer Architectures: A Design Space Approach", Addison Wesley, 1997.

    ii. M.J. Flynn, "Computer Architecture: Pipelined and Parallel Processor Design", Narosa Publishing House/ Jones

14. **Resources required for the course** Simulator Software

15. **Lecture outline with topics and number of lectures**

Topics	**Number of lectures**
General principles of performance enhancement	4

Instruction level parallel architectures	10
High performance memory systems	7
Shared memory multiprocessors	6
Multiprocessor interconnection networks	5
Performance modeling	3
Programming multiprocessors	3
Data parallel and multi-threading thread architectures	4

Course MCS-212 Compiler Design

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-212
3.	**L-T-P structure**	3-0-2
4.	**Credits**	4
5.	**Course Title**	Compiler Design
6.	**Prerequisites**	Theory of Computation
7.	**Status**	Core
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes

9.	**Frequency of offering**	EVEN semester ANNUAL

10. **Course objective**

To enable the students to appreciate importance of structuring and organizing large software with the help of theory and techniques learnt in other courses

11. **Course contents :** Overview of the Compiling Process, Some Typical Compiler Structures. Regular Expression, Finite Automata, Specification and Recognition of Tokens, Simple Approaches of Lexical Analyzer Design. Syntax trees, ambiguity, context tree grammar & derivation of parse trees, Basic parsing techniques, derivation, Top-Down and Bottom – Up Parsing, Operator – precedence Parsing, LR Parsers, Syntax Directed Definition, Translation schemes, L-attributed & S-attributed Definition. Data Structures For Symbol Tables (ST), Design of a ST for Block Structured Languages. Storage Allocation Strategies, Static Dynamic & Heap Memory Allocation, Memory Allocation in Block Structured Languages, In recursion, Memory allocation in Fortran Compilation of Expressions, Control Structures and I/O Statements, Error Detection and Recovery, Issues in Optimization, Optimizing Transformations, Local and Global Optimization, Loop optimization.

12. **Brief description of laboratory activities**

Implement a full compiler for a language, or implement various techniques for static analysis and type checking

13. **Suggested texts and reference materials**

i. Appel A W, Modern Compiler Implementation in ML, Cambridge University Press, 1997.

ii. Aho A V, Sethi R, Ullman J D, Compilers: Principles, Techniques, and Tools, Addison-Wesley 1986.

14. **Lecture outline with topics and number of lectures**

Topics	**Number of lectures**
Introductory Concepts	5
Lexical Analysis (Scanner)	5
Syntax Analysis (Parser)	10
Symbol Table Organization	4
Run-Time Memory Allocation	5
Compilation Process And Code Optimization	6

Course MCS-221 Computer Graphics

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-221

3.	**L-T-P structure**	3-0-6
4.	**Credits**	6
5.	**Course Title**	Computer Graphics
6.	**Prerequisites**	Data Structures
7.	**Status**	Major Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	EVEN semester ANNUAL

10. **Course objective**

To introduce the basic concepts of computer graphics.

11. **Course contents :** Graphics pipeline; Graphics hardware: Display devices, Input devices; Raster Graphics: line and circle drawing algorithms; Windowing and 2D/3D clipping: Cohen and Sutherland line clipping, Cyrus beck clipping method; 2D and 3D Geometrical Transformations: scaling, translation, rotation, reflection; Viewing Transformations: parallel and perspective projection; Curves and Surfaces: cubic splines, Bezier curves, B-splines, Parametric surfaces, Surface of revolution, Sweep surfaces, Fractal curves and surfaces; Hidden line/surface removal methods; illuminations model; shading: Gouraud, Phong; Introduction to Ray-tracing; Animation; Programming practices with standard graphics libraries like openGL.

12. **Brief description of laboratory activities**

Clipping, Viewing Pipeline, Hierarchical modeling using surfaces, Basic Ray Tracing.

13. **Suggested texts and reference materials**

i. Computer Graphics (Principles and Practice) by Foley, van Dam, Feiner and Hughes, Addisen Wesley (Indian Edition)

ii. Computer Graphics by D Hearn and P M Baker, Printice Hall of India (Indian Edition).

iii. Mathematical Elements for Computer Graphics by D F Rogers

14. **Lecture outline with topics and number of lectures**

Topics	**Number of lectures**
Graphics pipeline and hardware	4
Raster Graphics	3
2D and 3D Clipping	3
Transformations	6

Curves and Surfaces	10
Hidden Surface Elimination	5
Illumination Model and Shading	3
Ray Tracing	5
Animation	4

Course MCS-222 Operation Research

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course No**	MCS-222
3.	**L-T-P structure**	3-0-6
4.	**Credits**	6
5.	**Course Title**	Operation Research
6.	**Prerequisites**	Elementary Statistical theory and Methods
7.	**Status**	Major Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	EVEN semester ANNUAL

10. **Course objective**

To introduce the concepts of optimization Techniques

11. **Course contents**

Linear Programming Problems, Graphical Method and Simplex Method, Duality in LPP,Assignment and Transportation Problems, Degeneracy in Transportation Problem, Integer Programming, Branch and Bound Technique, Traveling Salesman Problem. Birth-death Process, Elements of Queuing Theory, M/M/1, M/M/K, Queuing Models with Infinite and finite Capacity, M/G/1 and G/M/1 Queuing Models, Priority Queues. Elements of PERT and CPM: Critical Path, Time Chart, Resource Levelling, Probability and Cost consideration in project scheduling Inventory System and ModelsKuhn Tucker Conditions, Quadratic Programming, Convex Programming, Sequencing Models: Classification of self problems, processing of n jobs through two, or three machines; Processing of two jobs through m machines

12. **Brief description of laboratory activities**

Programming assignments based on the above topics

13. **Suggested texts and reference materials**

B E Gillett: Introduction to Operations Research, A Computer Oriented Algorithmic Approach, Tata Mc Graw-Hill Publishing Co. Ltd, New Delhi, 1979

JG Ecker and M Kupferschmid: Introduction to Operations Reasearch, Joh Wiley & Sons,1988

Hamdy A Taha: Operations Research, Macmillan Publishing Co.Inc, 1982

M J Medhi: Stochastic Process, Wiley Eastern

JK Sharma: Mathematical Models in Operations Research Models, Tata Mc Graw-Hill Publishing Co. Ltd, New Delhi, 1982

FS Hillier& G J Lieberman:Introduction to Operations Research, Tata McGraw Hill Publishing Co. Ltd, New Delhi, 1982

14. **Resources required for the course** C, C++ compiler

15. **Lecture outline with topics and number of lectures**

Topics	**Number of lectures**
Linear Programming	12
Queuing Theory	8
Elements of PERT and CPM	5
Inventory System and Models	2
Kuhn Tucker Conditions, Quadratic Programming, Convex Programming	4
Sequencing Models	5

Course MCS-231 Neural Computing

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course No**	MCS-231
3.	**L-T-P structure**	3-0-0
4.	**Credits**	3
5.	**Course Title**	Neural Computing

6. **Prerequisites** -

7. **Status** Minor Elective

8. **Overlap with other UG/PG courses from other Departments/Centers** Yes

9. **Frequency of offering** EVEN semester ANNUAL

10. **Course objective**

To introduce the basic concepts and the trends in neural computing techniques

11. **Course contents**

Biological perspectives of neural network, Neural network modeling, Learning discriminants, Feedforward networks: perceptron, Multi layer networks, Complexity of learning using feedforward networks and adaptive structure networks. Case studies. Current trends in neural computing.

12. **Brief description of laboratory activities**

Design of various neural network architectures for given application and their performance evaluation

13. **Suggested texts and reference materials**

i. “Neural Networks in Computer Intelligence" by KM Fu, McGraw Hill, 1994

ii. “Neural Network Fundamentals with Graph, Algorithms and Applications” by N.K.Bose, Tata McGraw Hill, 1998

14. **Resources required for the course** Neural Networks Simulators

Course MCS-231 Bioinformatics

1. **Department proposing the Course** Department of Computer Science

2. **Course No** MCS-231

3. **L-T-P structure** 3-0-0

4. **Credits** 3

5. **Course Title** Bioinformatics

6.	**Prerequisites**	-
7.	**Status**	Minor Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	EVEN semester ANNUAL

10. **Course objective**

To introduce the basic concepts of Bioinformatics

11. **Course contents**

Introduction to Bioinformatics: Definition and History of Bioinformatics, Internet and Bioinformatics, Introduction to Data Mining, Applications of Data Mining to Bioinformatics Problems and Applications of Bioinformatics. Genomes: Introduction to nucleotides, amino acids, proteins, genes, introns, exons and their relationship; Introduction to organization of a genome: enzymes, operons, gene order, genome rearrangement, pathways and gene regulation. Human Genome Project: Influence Area and White Papers. Cloning and PCR, building genome maps and techniques for building genome maps, understanding of genomes from genome banks. Biological Databases: Overview of sequences: secondary and tertiary structures and metabolic pathway. Sequence Databases: Nucleotide Sequence, Protein Sequence, EMBL Neucleotide Sequence and Structure Data Bases. Bioinformatics Softwares: Clusal V, Cluster W 1.7, RasMol, Oligo, Molscript, Treeview, Alscript, Getic Analysis Software, Phylip. Biocomputing: Introduction to string matching algorithms, database search techniques, sequence Comparison and alignment techniques, use of biochemical scoring matrices, introduction to graph matching algorithms, Automated genome comparison and its implication, Automated gene prediction, Automated identification of bacterial operons and pathways. Introduction to signaling pathways and pathway regulation. Gene arrays, Analysis of gene arrays

12. **Suggested references**

i. Methods in Biotechnology and Bioengineering– SP Vyas & DV Kohli

ii. Exploring Genetic Mechanism– Maxine Singer & Paul Barg

iii. Evolutionary Computation in Bioinformatics– Gary B Fogel & David W Corne

iv. Genetic Library Construction and Screening: Advanced Techniques and Applications– Lab Manual

v. Techniques in Quantification and Localizationof Gene Expression– Bruce K Patterson

vi. Bioinformatics: Sequence and Genome Analysis– David W Mont

vii. Bioinformatics: Concepts, Skills and Applications– SC Rastogi and Namita Mendiratta

viii. Statistical Methods in Bioinformatics: An Introduction- WarrenJ evens, Gregory R Grant

ix. Statistical Genomics: Linkage Mapping and QTL Analysis- Ben Hui Liu

x. DNA Microarrays – David Bowtell & Joseph Sambrook

Course MCS-311 Analysis and Design of Algorithms

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-311
3.	**L-T-P structure**	3-0-6
4.	**Credits**	6
5.	**Course Title**	Analysis and Design of Algorithms
6.	**Prerequisites**	Data Structure
7.	**Status**	Core
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	ODD semester ANNUAL
10.	**Course objective**	

11. **Course contents**

Basic Computational Model and analyzing Algorithms, Asymptotic Notation and recurrence relations. Fundamental design methodologies and their implementations: Dynamics Programming, Greedy algorithms, Divide and Conquer, Branch and Bound, Backtracking, Randomized Techniques. Algorithms for set manipulations, their implementations and applications: Union-Find, Priority Queues. Graph Algorithms with implementation issues; Depth-First Search and its applications, minimum Spanning Trees and Shortest Paths. Matrix multiplication, Mattern Matching, polynomial arithmetic and FFT. Introduction to the Theory of Lower Bounds, NP-Completeness and Reductions

12. Brief description of laboratory activities

Laboratory work based on theory

13. Suggested texts and reference materials:

1. E. Horowitz & S. Sahani : Fundamental of Computer Algorithm (Galgotia)
2. Coreman, Leiserson & Rivest : Introduction to Algorithm (MIT)
3. Brassard & Brately : Algorithm- Theory and Practice (PHI)

Course MCS-312 Artificial Intelligence

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-312
3.	**L-T-P structure**	3-0-6
4.	**Credits**	4
5.	**Course Title**	Artificial Intelligence
6.	**Prerequisites**	Data Structures & Logic
7.	**Status**	Core
8.	**Overlap with other UG/PG courses from other Departments/Centers**	yes
9.	**Frequency of offering**	ODD Semester ANNUAL

10. **Course objective**

To expose students to search as a problem solving tool, knowledge representation using FOL, rules, frames, conceptual dependency, handling uncertainty, soft computing

11. **Course contents**

Problem solving, search techniques, control strategies, game playing (minimax), reasoning, knowledge representation through predicate logic, rule based systems, semantics nets, frames, conceptual dependency formalism. Planning. Handling uncertainty: Bayesian Networks, Dempster-Shafer theory, certainty factors. Fuzzy logic, Learning through Neural nets - Back propagation, radial basis functions, Neural computational models - Hopfield Nets, Boltzman machines. PROLOG programming. Expert Systems.

12. **Brief description of laboratory activities**

. Programming assignments involving implementation using languages such as PROLOG, LISP will be given.

13. **Suggested texts and reference materials**

i. “AI, a modern approach" by Russel and Norvig, Pearson Education

ii. "AI" by Rich and Knight, Tata McGraw Hill

iii. "Neural Networks in Computer Intelligence" by KM Fu, McGraw Hill

14.	**Resources required for the course**	PROLOG, LISP, FXCLIPS, Neural Networks Simulators

15. **Lecture outline with topics and number of lectures**

Topics	**Number of lectures**
What is an AI technique?	2
State space search, control strategies	3
Heuristic Search, Best-first, A*, constraint satisfaction	4
Game playing- alpha beta cut off	3
Knowledge representation, Planning	5 + 3
Bayesian networks, Dampster Shafer, certainty factors	6
PROLOG	3
Fuzzy logic, Neural networks, Expert systems	4 + 6 + 3

Course MCS-313 Independent Study

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-313
3.	**L-T-P structure**	0-3-0
4.	**Credits**	3
5.	**Course Title**	Independent Study
6.	**Prerequisites**	-
7.	**Status**	Core
8.	**Overlap with other UG/PG courses from other Departments/Centers**	No

9. **Frequency of offering** ODD semester ANNUAL

10. **Course objective**

To develop independent research abilities in students on material outside regular courses

11. **Course contents**

Research oriented activities or study of advanced subjects outside regular course offerings under the guidance of a faculty member. Prior to registration, a detailed plan of work should be submitted by the student in concurrence with a faculty guide.

Course MCS-321 Simulation and Modeling

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course No**	MCS-321
3.	**L-T-P structure**	3-0-6
4.	**Credits**	6
5.	**Course Title**	Simulation and Modeling
6.	**Prerequisites**	
7.	**Status**	Major Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	ODD semester ANNUAL

10. **Course objective**

To introduce the basic concepts of Simulation and Modeling

11. **Course contents**

Simulation and its uses, Definition of System, Types of Systems, Steps of Simulation Process. Simulation Experiments and Field Experiments Concepts of Random Sequences, Random Number Generators from Uniform and other Continuous and Discrete Distributions Tests of Randomness and Goodness of Fit Modeling Process and Concepts of Mathematical Models Differential, Partial Differential and Difference Equation Models, Modeling through Graphs, Stochastic Models, Ethernet Model, Monte-Carlo Integration Simulation of Single Server System, Inventory System, Time Sharing Computer System, and Ethernet Model. Verification, Validation and Comparison of Real System and Simulation Experiment Data, Variance Reduction Techniques and Sensitivity Analysis Simulation Languages: SIMULA, SIMSCRIPT and GPSS

12. **Brief description of laboratory activities**

Programming assignments on Simulation and Modeling

13. **Suggested texts and reference materials**

i. JA Payne Introduction to Simulation, Programming Techniques and Methods of Analysis, Tata McGraw Hill Publishing Co. Ltd, 1988

ii. AM Law & WD Kelton: Simulation Moelling & Analysis, McGraw Hill Inc. 1991

iii. MH MacDougall: Simulating Computer Systems: Techniques& Tools, The MIT Press Cambridge, 1987

iv. ZA Klarian & EJ Dudewicz: Modern Statistical Systems and GPSS Simulation, Computer Science Press1990

v. G Gordon: System Simulation, PHI, 1995

vi. Narsingh Deo:System Simulation with Digital Computer, PHI, 1997

vii. JN Kapoor: Mathematical Modelling, Wiley Eastern Ltd. 1988

viii. BP Zeigler, H Praehofer, TG Kim: Theory of Modelling and Simulation-Integrating Discrete Event and Continuous Complex Dynamic Systems, Academic Press 2000

14. **Resources required for the course**

-

15. **Lecture outline with topics and number of lectures**

Topics	**Number of lectures**
Simulation and its uses, Definition of System	5
Concepts of Random Sequences	5
Tests of Randomness and Goodness of Fit	3
Modeling	2
Differential, Partial Differential and Difference Equation Models, Modeling through Graphs, Stochastic Models,	10

Ethernet Model, Monte-Carlo Integration	5
Simulation of Single Server System, Inventory System, Time Sharing Computer System, and Ethernet Model	3
Verification, Validation and Comparison of Real System etc	5
Simulation Languages	3

Course MCS-331 Parallel Computing

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course No**	MCS-331
3.	**L-T-P structure**	3-0-0
4.	**Credits**	3
5.	**Course Title**	Parallel Computing
6.	**Prerequisites**	Advanced Architecture
7.	**Status**	Minor Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	ODD semester ANNUAL

10. **Course objective**

This course is aimed at providing students with a deep knowledge of the techniques and tools needed to understand today's and tomorrow's high performance computers, and to efficiently program them

11. **Course contents**

Review of Multiprocessor and distributed systems. Conditions of parallelism, program partitioning and program flow mechanism. Parallel models: Shared memory model, message passing model, data parallel model, object-oriented model, functional and logic models. Parallel Language and Compilers: Language features for parallelism, parallel language constructs, optimizing compilers for parallelism, dependency analysis, code optimization and scheduling, loop parallelization and pipelining. Parallel Program development: parallel programming environments, synchronization and multiprocessing modes, shared variable program structures, message passing, program development mapping programs onto multiprocessors. Multiprocessor UNIX (design goals), Master slave and multithreaded unix, multicomputer unix extension, Mach/OS kernel architecture, OSF/1 architecture and programming environment.

12. **Suggested text and reference material**

i. Kai Hwang and Zhiwei Xu, Scalable Parallel Computing, McGraw Hill New York, 1997.

Course MCS-331 Distributed Computing

1.	**Department proposing the Course**	Department of Computer Science
2.	**Course Number**	MCS-331
3.	**L-T-P structure**	3-0-0
4.	**Credits**	3
5.	**Course Title**	Distributed Computing
6.	**Prerequisites**	Operating Systems, Advanced Architecture.
7.	**Status**	Minor Elective
8.	**Overlap with other UG/PG courses from other Departments/Centers**	Yes
9.	**Frequency of offering**	ODD semester ANNUAL

10. **Course objective**

To introduce the practical distributed computing problems within the framework of abstract mathematical models, and develop the tools and techniques using which non-trivial properties can be identified, their complexity assessed and distributed systems specified.

11. **Course contents**

Models of Distributed Computing; Basic Issues: Causality, Exclusion, Fairness, Independence, Consistency; Specification of Distributed Systems: Transition systems, petri nets, process algebra properties: Safety, Liveness, and stability.

12. **Books :**

1. Hwang & Briggs : Computer Architecture & Parallel Processing (McGraw Hill)
2. Crich Low : In to Distributed & Parallel Computing (PHI)
3. V. Rajaraman : Element of Parallel Computing (PHI)
4. Nancy A. Lynch : Distributed Algorithms, Morgan Kaufmann 1996
5. Gerald Tel. : Introduction to Distributed Algorithms. CUP.