M. SC. INFORMATION TECHNOLOGY – FINAL
(2011– 2012)

Paper Name (Theory)	Lec	Tut	Exam Hours	Max Marks	
				Sess Marks	Exam marks
MIT – 201 Operating Systems	3	1	3	20	80
MIT – 202 Object Oriented Concept & C++	3	1	3	20	80
MIT – 203 Computer Oriented Numerical Methods	3	1	3	20	80
MIT – 204 Software Engineering	3	1	3	20	80
MIT – 205 Programming in Java	3	1	3	20	80
MIT – 206 Artificial Intelligence	3	1	3	20	80
MIT – 207 Cyber Law & Internet Security	3	1	3	20	80
MIT – 208 Software Testing & Quality Assurance	3	1	3	20	80
Total of Theory					**800**

Paper Name (Practical)	Pract Hours	Pract Exam	Min Marks	Max Marks
MIT – 209 Programming Lab – 5 (C++ & Software Engg)	3	3	20	50
MIT – 210 Programming Lab – 6 (Unix, Shell Prog and Numerical Programming in C)	3	3	20	50
MIT – 211 Programming Lab – 7 (Prolog Programming)	3	3	20	50
MIT – 212 Programming Lab – 8 (Java, Exploring Internet)	3	3	20	50
MIT – 213 Project	6	3	40	100
Total of Practical				300
Grand Total (Theory + Practical)				**1100**

The question paper will be divided into 3 parts:

Part A:

1. 10 Question of 1 mark each
2. Answer should not exceed more than 20 words
3. All questions are compulsory

Part B:

1. 5 Questions of 2 marks each
2. Answer should not exceed more than 50 words
3. All questions are compulsory

Part C:

1. 3 Questions of 20 marks each. There will be an internal choice in each question.
2. Answer should not exceed 400 words
3. All questions are compulsory.

Note:

1. **Lec:** Lectures **Tut:** Tutorials per week per hour
2. There will be sessional (internal assessment) of 20 marks conducted by the college
3. Two Practical exams shall be conducted by one internal and one external examiner of a batch of 20 students in day.
4. **Project Work:** 6 hours per student
   The Project Report work shall be assessed by one internal and one external examiner only of a batch of 20 students in a day. The project work should not be

done in a group. Each student shall be allotted one project and one copy should be submitted to the University.

5. Duration of Practical exam is 3 hours.
6. A Laboratory Exercise File should be prepared by each student for each practical paper and should be submitted during practical examinations.
7. Practical of 50 marks distribution is as under:
    a. 30 marks for practical examination exercise for 3 questions
    b. 10 marks for Viva-voce
    c. 10 marks for Laboratory Exercise File
8. Eligibility: Graduate from any discipline with 50% marks in aggregate. As regards admission on reserved category seats government rules will be applicable

Scheme of Examination
(For M. Sc. Computer Science/Information Technology)

Reg. 17 (a)

The examination for the M. Sc. Computer Science/Information Technology will consist of (a) Theory papers (b) Laboratory / Practical work and project work. Candidates will be required to pursue a regular, full time course of study at the University department and the affiliated colleges for a period of two academic years in order to be eligible for appearing in the examination.

1. Eligibility for M. Sc. Computer Science/Information Technology: 50% marks in any graduation scheme.
2. Examination:
    i. There shall be 25 papers (16 theory, 8 practical in each year and 1 project as practical in the final year) of 2100 marks (previous and final year). Theory paper shall be of 3 hours duration having 100 marks. Out of 100 marks 20 marks shall be considered as internal assessment based on internal test and seminars and 80 marks will be of examination determined by the University. The practical shall be of 50 marks assessed by external examiner and the project work shall be of 100 marks based on project presentation and viva-voce, assessed by external examiner.
    ii. For passing a candidate shall have to secure at least 40% marks in each course (theory paper, sessional and practical work separately) and 50% marks in the aggregate in all the courses.
    iii. Due paper(s) will be applicable if a candidate obtains 50% marks in aggregate and fails in not more than three (3) papers (theory). Due paper(s) will be held along with the examination of the next year. The chance of due paper(s) will be given only 2 times.
    iv. Wherever a candidate appears at for a due paper examination he/she will do so according to the syllabus in force.
    v. A candidate not appearing at any examination/absent in any paper of term end examination shall be deemed as fail.

3. A candidate for a pass in the examination shall be required to obtain:
    i. At least 50% marks in the aggregate of all the papers prescribed for the examination and
    ii. At least 50% marks in the practical(s) wherever prescribed at the examination, provided that if a candidate fails to secure at least 40% marks in each individual paper at the examination notwithstanding his having obtained the minimum percentage of marks required in the aggregate for that examination.

    No Division will be awarded in the Previous Year examinations. Division shall be awarded at the end of the Final Year Examination on the combined marks obtained at the previous and final examinations taken together as noted below:

Passed with First Division	60% of the aggregate marks taken together of previous and final examinations
Passed with second division	48%

    Provided that if a candidate clears any paper after a continuous period of two years since he/she was admitted to the M. Sc. Computer Science/Information Technology then for the passing marks, i.e. 40% marks, shall be taken into account in the case of such course(s).

    Provided further that in case where a candidate requires more than 40% marks in order to reach the requisite minimum aggregate as many marks, out of those actually secured by him/her will be taken into account as would enable him/her to make up the deficiency in the requisite minimum aggregate marks.

4. Candidates reappearing at an examination in a subsequent year shall be examined in accordance with the scheme and syllabi in force and shall be entitled to the award of the degree of year in which they clear the last failing/unclear paper.

MIT – 201 OPERATING SYSTEMS

Introduction to Operating Systems, goals of OS, operation of OS, resource allocater and related functions, classes of OS, batch processing, multi-processing, time sharing, distributed, real time systems, system calls, system programs, structure of OS, layer design of DOS, Unix,

Process concept, interacting process, threads, process and thread in Windows 2000, process scheduling, fundamental of scheduling, scheduling criteria, long medium short term scheduling, scheduling algorithms upto multi-processor scheduling, algorithm evaluation, critical section, critical region, inter-process communication, monitor and semaphores, implementation and uses.

Logical versus physical address, swapping, contiguous allocation, segmentation, paging, segmentation with paging, kernel memory allocation, page replacement algorithm, virtual memory, virtual memory with paging, demand paging, dead lock, characterization, methods for handling dead locks, prevention, avoidance, thrashing, allocation of frame, virtual memory using segmentation, Windows 2000.

History of Linux, Linux architecture, Linux File System, file naming, types of files, directory command, file command, vi editor, locating files in Linux, filter, pipe, shell variables, local and global variables, command substitution, if, while, for, shift, tar, basic networking commands in Linux.

Reference:

1. Operating System Linux, NIIT PHI
2. Operating System Concepts, Galvin, Addison Wesley
3. Operating Systems, Ritchie, BPB Publications.

MIT – 202 OBJECT ORIENTED CONCEPTS AND C++

Data types, operator, input-output, control statements, loops, arrays, strings and string functions, functions, structure and union, Introduction to OOPS, object oriented analysis and design, class, declaring object, member function, data hiding, parameter passing, friend function and class, empty static, overloading, constructor, type of constructor, destructor, recursive constructor, calling constructor and destructor, overloading unary operator, binary operator with friend function, rule of overloading.

Inheritance, derive and base class, overriding, base and derive constructor, type of inheritance, virtual base class, abstract class, qualifier class and inheritance, pointer, pointer to class, pointer to object, pointer to derived class and base class, pointer to member, pointer to array, accessing private member and direct access to private member, new delete operator, dynamic memory,

Binding in C++, virtual function, rule for virtual function, pointer to derive class object, pure virtual function, constructor and virtual functions, polymorphism, file, file operator and commands, use in C++, templates.

Single linked lists, double linked list, circular list, sparse table, stack, queue, d-queue list, priority queue, graph, prefix, postfix, infix, sorting, insertion, selection, bubble, algorithm of quick, merge, radix, heap.

Searching, binary, linear, tree, Binary Tree, Binary search tree, tree traversal, breadth – first, depth – first, AVL tree, B-tree, in-order, pre-order, post-order graph, BFS, DFS, shortest path, algorithm of Kruskal, prim.

Reference Books:

1. Object Oriented Programming with C++, E. Balagurusamy, Tata McGraw Hill.
2. Data Structures and algorithms in C++, Adam Drozdex, Vikas Publications
3. Understanding Programming an introduction using C++, Scott R Canon, Vikas Publications.
4. OOPS with C++, N P Bhave,
5. OOPS with ANSI C++, A N Kamthane,

MIT – 203 COMPUTER ORIENTED NUMERICAL METHODS

Characteristics of Numerical Computation, Approximation, Significant Digit, Errors, Introduction to Matrix, Types of Matrix, Square, Row, Column, Diagonal, Unit, Null, Upper Triangular, Lower Triangular, Symmetric, Skew Symmetric, operation of matrix, trace, transpose, addition, subtraction, multiplication, determinant, inverse, Introduction to Linear Equations,

Bisection method, method of successive approximation, method of false position, Newton's iteration method, Newton Raphson method, Horner's method

Gauss Jordan method, Gauss Elimination method, Iterative methods, Jacobi method of iteration, Gauss Seidel Iteration method

Finite differences, forward, backward, central differences, other difference operations and its relations, difference of polynomial, fractional polynomial, reciprocal factorial

Gregory Newton Forward and Backward interpolation Formula, Gauss Forward and backward difference interpolation formula, interpolation with unequal intervals,

Euler methods, modified Euler Method, Runge Kutta 2^{nd} and 4^{th} order method, trapezoidal rules, Simpson's 1/3, 3/8 rules, Weddle's rule.

Reference:

1. Numerical Methods, Dr. V. N. Vedamurthy & Iyengar Vikas Publication
2. Computer Oriented Numerical Methods, V.Rajaraman
3. Computer Oriented Numerical Methods, Iyengar & Iyengar

MIT – 204 SOFTWARE ENGINEERING

Concepts of Software Engineering, Software Characteristics, components applications, software Metrics and Models; Process and Product Metrics, Size metric, Complexity metric, McCabe's Cyclometic Complexity, Halsted Theory, Function Point Analysis.

System Development Life Cycle (SDLC) Steps, Water fall model, Prototypes, Spiral model. Planning and Software Project: Cost Estimation, Project Scheduling, Quality Assurance Plans, project Monitoring Plans.

Software Development & Software Design : System design, detailed design, function oriented design, object oriented design user Interface design, Design level metrics: Phases, Process Models, Role of Management, Role of Metrics and Measurement, Software Quality factors,

Coding and Testing: Programming Practices, verification, Monitoring and Control. Testing level metrics Software quality and reliability Clean room approach, software reengineering.

Testing & Reliability: Testing Fundamentals, Test case design, Functional Testing, Structural Testing, Test Plan activities during testing, Unit System , Integration Testing. Concept of Software Reliability, Software Repair and Availability, Software Errors and Faults Reliability Models (JM, GO, MUSA Markov) Limitations of Reliability Models

Reference:

1. Software Engineering Fundamentals, Ali Behforooz, Oxford Univ Press.
2. Software Engineering, Pressman, R. S. Pressman & Associates.
3. Software Engineering, Sommerville, Addison Wesley

MIT – 205 PROGRAMMING IN JAVA

Introduction to Java, history, characteristics, Object oriented programming, data types, variables, arrays, difference between Java and C++

Control statements: selection, iteration, jump statements, operators

Introduction to classes, class fundamentals, constructor, methods, stack class, inheritance, creating multilevel hierarchy, method over riding

Packages and interfaces, exception handling, multi-threaded programming, I/O applets

Java Library, string handling, string comparison, string buffer, utility classes, vector stack dictionary, applet class, introduction to AWT, working with frame windows.

Java beans, beans architecture, AWT components, advantage of Java beans, beans serialization, JDBC, class and methods, API components, JDBC components, driver, connectivity to database, processing result and interfaces, RMI, comparison of distributed and non-distributed Java programs, interfaces, RMI architecture layer, ODBC, CORBA, CORBA services and products, CGI, structure of CGI.

Reference books:-

1. Y. Daniel Liang, Introduction to Java Programming, PHI.
2. Patrick Naughton, Java Complete Reference, Tata McGraw Hill.
3. The Java Handbook, Patrick Naughton, Tata McGraw Hill.
4. Introduction to Java Programming, E Balaguruswamy, PHI.
5. Programming Java, Decker & Hirshfield, Vikas Publications

MIT – 206 ARTIFICIAL INTELLIGENCE

Definition of AI, Application of AI, knowledge-based system, representation of knowledge, organization, manipulation and acquisition of knowledge,

Introduction of prolog, variable, object, domain, clauses, recursion, basic list manipulation function, predicates, input, output, local variable, iteration, recursion, arrays, database in prolog, rule order, goal order, cut, trial, prolog query.

Syntax, semantics of propositional logic, syntax and semantics of FOPL, conversion to clausal form, inference rule, resolution principles, non-deductive inference methods, representation using rules, truth maintenance system, predicate completion and circumscription, modal and temporal logics, fuzzy logic,

Bayesian probabilistic inference, possible word representations, Dempster-Shafer Theory, Ad-Hoc methods, Heuristic reasoning methods, associative networks, frame networks, search problems, uniformed or blind search, searching And-Or graph

Matching techniques, measures for matching, matching like patterns, partial matching, Fuzzy matching algorithms, indexing and retrieval techniques, integrating knowledge and memory

Expert system, rule based system architecture, non-productive system architecture dealing with uncertainty, knowledge acquisition and validation, knowledge system building tool.

Reference Books:

1. Artificial Intelligence and Expert System, D. W. Patterson, PHI
2. Artificial Intelligence, Ritchie

MIT – 207 CYBER LAW & INTERNET SECURITY

Introduction: Issues in Network Security, Network Security Services, basic Concepts of Encryption and Decryption, Substitution Ciphers, Transposition Ciphers. Electronic Mail Security, IP Security, WEB Security, Intruders, Viruses and Worms, Firewalls.

Cyber Laws: Cyber laws for Cyberspace- Legal Identity and Private International Laws in Cyberspace. E-Commerce, E-Commerce, Issues of Privacy

The World of Electronic Contracts - E-Agreements and the Web Surfing, Terms of Service Contracts, Terms of Service Agreement for Web Site Owners, Tips to Frame a Private Policy for and E-commerce Site

Cyber Pirates - Copyright, Digital Content right. Steps to protect the Contents on WWW, Software Patents, Domain Name System and Trademarks Crimes-Cyber Crimes and Future Imperfect, Strategy to Combat Cyber Crimes,

Cryptography: Basic Terms and Concepts, Brief History of Cryptography and Cryptanalysis. Uses and misuses. Basic Number Theory - Divisibility, Primarily, Bases, Congruence's, Modular Arithmetic, GCD's, Euclidian algorithm, Fermat and Euler Theorems, Finding large primes, Pohlig-Hellman, RSA.

Reference Books:

1. William Stallings, "Cryptography and Network Security : Principles and Practice", Pearson Education, 2000.
2. Kernal Texpalan, "Communication network Management:, PHI, 1992.
3. D.E. Cormer, "Computer Networks and Internet", 2nd Edition, Addison Wesley Publication, 2000.

MIT – 208 SOFTWARE TESTING & QUALITY ASSURANCE

Need for Testing—Psychology of testing—Testing economics—white box testing , Black box testing, Grey box Testing—Retesting regression Testing—Verification and Validation Testing Strategies—Levels of Testing—Unit, Integration ,System Testing, Acceptance Testing

Test case Design—Statement Coverage—Branch Coverage—Condition Coverage—Decision / Condition Coverage—Multiple Condition Coverage—Data Flow Coverage—Mutation Testing

Test Case Designs. Boundary Value analysis—Equivalence Partitioning—Cause Effect Graphing, Error Guessing, Logic Based Testing.

Special Topics: Syntax testing—Finite State Testing Logic Based Testing Domain Testing

Test Planning—Test Plan Documentation—Test Estimation—Test Schedule —Test monitoring and Control—standards for Testing.

Introduction of Object Oriented Testing—Automated Tools for Testing—Tool Selection and Implementation—Test case generators—GUI Testing—Testing Web enabled Application.

Reference books:

1. Glenford J.Myers," The Art of Software Testing" John Wesley & Sons 1979.
2. Boris Beizer, "Software Testing Technologies" 1st edition Dreamtech 2000.
3. Roger S.Pressman," Software Engineering" 5th edition, Mc Graw Hill
4. William E.Lewis," Software Testing and continuous quality improvement "Auerbach