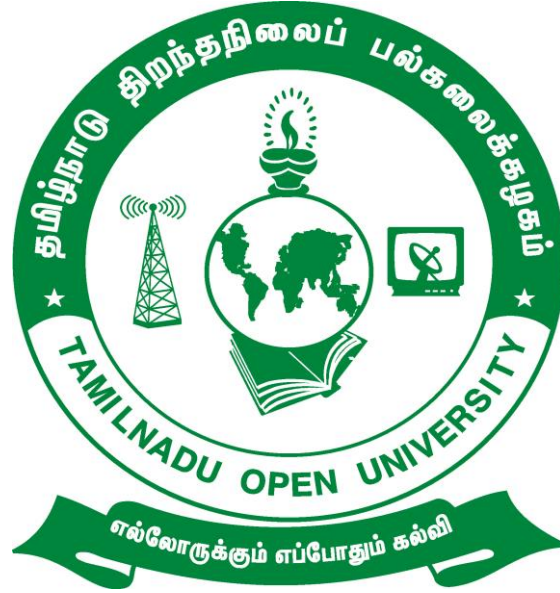


Tamil Nadu Open University

Regulations and Overview for Master of Science in Computer Science (Semester)

[w.e.f. Academic Year 2020-2021]



School of Computer Science
Tamil Nadu Open University
Chennai - 600 015.

MASTER OF SCIENCE IN COMPUTER SCIENCE: M.SC-CS

REGULATIONS

1. Programme Objectives:

- To train the students to understand the basics of Computer Science.
- To improve the ability to solve problems, think independently and to understand the role of Computer Science in Information technology.
- To realize the significance of software development.
- To use appropriate theory, practices and tools to design, implement, test and evaluate systems to solve problems in Computer Science and other fields.
- To provide Employment Opportunity to the Unemployed in Rural / Urban areas.

2. Programme Outcomes (POs):

- Scientific knowledge: Apply the knowledge of mathematics, science, and computing to the solution of complex scientific problems.
- Problem analysis: Identify, formulate, research literature, and analyze complex scientific problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and applied sciences.
- Design/development of solutions: Design solutions for complex problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- Modern tools usage: Create, select, and apply appropriate techniques, resources, and modern computing and IT tools including prediction and modeling to complex scientific activities with an understanding of the limitations.
- The software engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional practice.
- Environment and sustainability: Understand the impact of the professional software engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the scientific practice.

- Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- Communication: Communicate effectively on complex activities with the scientific community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- Project management: Demonstrate knowledge understanding of the scientific and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

3. Programme Specific outcomes (PSOs):

A graduate with a M.Sc. in Computer Science will have the ability to

- Communicate computer science concepts, designs, and solutions effectively and professionally
- Apply knowledge of computing to produce effective designs and solutions for specific problems
- Use software development tools, software systems, and modern computing platforms

4. Eligibility:

B.Sc. Computer Science / B.C.A. /B.Sc. Computer Technology / B.Sc. Information Technology /B.Sc Information Sciences/B.Sc Information Systems/ B.Sc. Software Systems/B.Sc Software Sciences / B.Sc Applied Sciences (Computer Science/Computer Technology) and hitherto equivalent degrees.

5. Duration of the Course:

The course for the degree of Master of Science (Computer Science) shall consist of two academic years (Four Semester).

5. Medium : English

6. Mode of Study: Distance Mode

7. Admission:

The candidate's admission for the degree of Master of Science in Computer Science will be taken in Academic year and Calendar year.

8. Course of Study:

The course of study shall comprise instruction in the following subjects according to the syllabus

First Year	
First Semester	<ol style="list-style-type: none">1. Mathematical Foundations of Computer Science2. Data Structures3. Distributed Operating System4. Advanced Database Management Systems5. Object Oriented Programming Using C++6. Data Structure Using C++ Laboratory7. Advanced Database Management Systems Laboratory
Second Semester	<ol style="list-style-type: none">1. Analysis of Algorithm2. Cloud Computing3. Internet of Things4. Data Mining5. Application Programming for Mobile Devices6. Application Programming for Mobile Devices Laboratory7. Algorithm Laboratory
Second Year	
Third Semester	<ol style="list-style-type: none">1. Web Technology2. Network Security3. Big Data4. Elective I5. Elective II6. Web Technology Laboratory7. Mini Project (Using Open Source Software)
Fourth Semester	<ol style="list-style-type: none">1. Elective III (CBCS)2. Elective IV (CBCS)3. Project Work

9. Examinations:

The examination for the M.Sc. Degree shall consist of theory and practical papers.

(i) Theory Examinations: The theory examination shall be of three hours duration to each paper and conducted at the end of each year. The candidates who failing in any subject(s) will be permitted to appear for each failed subject(s) in the subsequent examinations.

(ii) Practical Examinations: The practical examinations shall be of three hours duration to each practical and conducted at the end of each year. The candidates who failing in any practical(s) will be permitted to appear for each failed practical(s) in the subsequent examinations.

(iii) Project Examinations: The project examinations shall be of three hours duration to each the project and conducted at the end of each year. The candidates who failing in the project examinations will be permitted to appear in the subsequent examinations.

(Note: Those who fail in the practical examination should appear only for practical examinations not need to appear for the practical counselling classes again. The marks once awarded for records will remain the same and will be declared to have passed in that course.)

10. Scheme of Examinations:

S.No.	Course Code	Course Name	Category	TEE + CIA	Total Marks	Min. Pass Mark	Credits
First Semester							
1.	MSCS-1	Mathematical Foundations of Computer Science	CC	70+30	100	50	4
2.	MSCS-2	Data Structures	CC	70+30	100	50	4
3.	MSCS-3	Distributed Operating System	CC	70+30	100	50	4
4.	MSCS-4	Advanced Database Management Systems	CC	70+30	100	50	4
5.	MSCS-5	Object Oriented Programming Using C++	CC	70+30	100	50	4
6.	MSCS P1	Data Structure Using C++ Laboratory	SEC	70+30	100	50	2
7.	MSCS P2	Advanced Database Management Systems Laboratory	SEC	70+30	100	50	2
Second Semester							
8.	MSCS-6	Analysis of Algorithm	CC	70+30	100	50	4
9.	MSCS-7	Cloud Computing	CC	70+30	100	50	4
10.	MSCS-8	Internet of Things	CC	70+30	100	50	4
11.	MSCS-9	Data Mining	CC	70+30	100	50	4
12.	MSCS-10	Application Programming for Mobile Devices	CC	70+30		50	4
13.	MSCS P3	Application Programming for Mobile Devices Laboratory	SEC	70+30	100	50	2
14.	MSCS P4	Algorithm Laboratory	SEC	70+30	100	50	2

Third Semester							
15.	MSCS-11	Web Technology	CC	70+30	100	50	4
16.	MSCS-12	Network Security	CC	70+30	100	50	4
17.	MSCS-13	Big Data	CC	70+30	100	50	4
18.		Elective I	DSE	70+30	100	50	4
19.		Elective II	DSE	70+30	100	50	4
20.	MSCS P5	Web Technology Laboratory	SEC	70+30	100	50	2
21.	MSCS P6	Mini Project (Using Open Source Software)	DSE	70+30	100	50	2
Fourth Semester							
22.		Elective III (CBCS)	GE	70+30	100	50	4
23.		Elective IV (CBCS)	GE	70+30	100	50	4
24.	MSCS P7	Project Work	DSE	-	200	100	8
Total credits							88

* CC- Core Course,

DSE- Discipline Specific,

SEC – Skill Enhancement Course,

GE- Generic Elective,

CIA- Continuous Internal Assessment

TEE- Term End Examination

LIST OF ELECTIVES

Elective I		
1.	MSCS-14	Artificial Intelligence
2.	MSCS-15	Mobile Computing
3.	MSCS-16	User Interface Design
4.	MSCS-17	Unix Internals
5.	MSCS-18	Big Data Query Languages
Elective II		
1.	MSCS-19	Software Testing
2.	MSCS-20	Supply Chain Management
3.	MSCS-21	Block chain Technologies
4.	MSCS-23	Software Project Management
5.	MSCS-23	Software Quality Assurance

Elective III	
1	Any one PG Course Offered from Other Departments of Tamil Nadu Open University.
ELECTIVE – IV	
1	Any one PG Course Offered from Other Departments of Tamil Nadu Open University.

11. Question Pattern:

Tamil Nadu Open University
Examination Section – SRE-1
Term End Examination _____
M.A. / M.Sc. / MBA / MCA Degree Examination
(Batch CY 2020 Onwards)

Course : XXXX

Course Code: XXXX

Time : 3 hours

Maximum marks : 70

PART - A ($5 \times 5 = 25$ marks)

Answer any Five questions out of Eight Questions in 300 words

All questions carry equal marks

1. From Block 1 - I
2. From Block 1 - II
3. From Block 1 - III
4. From Block 1 - IV
5. From Block 1 - V
6. From any Block 1
7. From any Block 1
8. From any Block 1

PART - B ($3 \times 15 = 45$ marks)

Answer any Three questions out of Five Questions in 1000 words.

All questions carry equal marks.

9. From Block 1 - I
10. From Block 1 - II
11. From Block 1 - III
12. From Block 1 - IV
13. From Block 1 - V

12. Passing Minimum:

i) For theory examination: The candidate shall be declared to have passed the examination if the candidate secures not less than 32 marks in the Term End Examinations (TEE) of each theory paper and secures not less than 13 marks in the Continuous Internal Assessment(CIA)and overall aggregated marks is 50 in both the external and internal taken together.

Continuous Internal Assessment (CIA)			Term End Examination (TEE)		Overall Aggregated Marks	Maximum Marks
Minimum Mark	Pass Mark	Maximum Mark	Minimum Mark	Maximum Mark	CIA + TEE	
13		30	32	70	50	100

ii) For practical examination: The candidate shall be declared to have passed the examination if the candidate secures not less than 30 marks in the External Practical Examinations and secures not less than 10 marks in the Continuous Internal Assessment (CIA) (Record Marks + Practical Counselling, Class Attendance) and overall aggregated marks is 40 marks in both external and internal taken together. However submission of record notebook is a must.

iii) Mini Project: the following pattern for mini project work for Master of Science in Computer Science M.Sc.,(CS): The mini project work will be evaluated for 100 marks. The 100 marks is awarded for Evaluation of the Project. The passing minimum in the mini project work is 50% of the total mark. i.e. the student should get minimum 50 marks out of 100 marks in the mini project evaluation and the viva-voce examination.

iv) Project: the following pattern for project work for Master of Science in Computer Science M.Sc.,(CS) Full-time: The project work will be evaluated for 200 marks. The 150 marks is awarded for Evaluation of the Project and 50 marks for Viva-voce examination. The passing minimum in the project work is 50% of the total mark. i.e. the student should get minimum 100 marks out of 200 marks in the project evaluation and the viva-voce examination.

13. Awarding of marks for Practical examinations.

The following pattern is prescribed for practical courses for Master of Science in Computer Science M.Sc.,(CS) in distance mode

S.NO	Activities	Marks
1	Writing the Program	20
2	Compiling Program	20

3	Running Program	10
4	Record Work	10
5	Viva	10
Total		70

13. Awarding of marks for Mini Project examinations.

SNO	Activities	Marks
1	Analysis and Design	50
2	Coding	25
3	Organization of Report	25
Total		100

15. Awarding of marks for Project examinations.

SNO	Activities	Marks
1	Analysis and Design	50
2	Coding	50
3	Organization of Report	50
5	Viva - voce	50
Total		200

16. Classification of Successful Candidates:

Candidates who pass all the courses prescribed and who secure 60% and above in the aggregate of marks in Core courses will be placed in the First Class. Those securing 50% and above but below 60% in the aggregated will be placed in the Second Class. All other successful candidates will be placed in the Third Class.



Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE	:	Mathematical Foundations of Computer Science
COURSE CODE	:	MSCS-1
COURSE CREDIT	:	04

COURSE OBJECTIVES

While studying the Mathematical Foundations of Computer Science, the student shall be able to:

- Introduces formal logic and its connections to Computer Science. Students learn to translate statements about the behavior of computer programs into logical claims and to prove such assertions both by hand and using automated tools.
-

COURSE OUTCOMES

After completion of the Mathematical Foundations of Computer Science, the student will be able to:

- To apply knowledge of computing and mathematics appropriate to the discipline.
 - To identify, formulate, and develop solutions to computational challenges.
 - To design, implement, and evaluate a computational system
 - To meet desired needs within realistic constraints.
-

Block –1:

Formal Logic: Statements, Symbolic Representation, and Tautologies, Propositional Logic, Quantifiers, Predicates, and Validity, Predicate Logic, Logic Programming, Proof of Correctness. Proofs, Recursion.

Block -2:

Sets, Combinatorics, Probability, and Number Theory: Sets, Counting, Principle of Inclusion and Exclusion; Pigeonhole Principle, Permutations and Combinations, Probability, Binomial Theorem, Number Theory.

Block -3

Relations, Functions, and Matrices: Relations, Topological Sorting, Relations and Databases, Functions, The Mighty Mod Function, Matrices, Recursive Function-Recurrence relations.

Block -4

Graphs and Trees: Graphs and their Representations, Trees and their Representations, Decision Trees, Huffman Codes. Graph Algorithms: Directed Graphs and Binary Relations; Warshall's Algorithm, Euler Path and Hamiltonian Circuit, Shortest Path and Minimal Spanning Tree, Traversal Algorithms, Articulation Points and Computer Networks.

Block -5

Boolean Algebra and Computer Logic: Boolean Algebra Structure, Logic Networks, Minimization, Modeling Arithmetic, Computation, and Languages: Algebraic Structures, Finite-State Machines, Turing Machines, Formal Languages.

Reference book:

1. Judith L.Gersting, “Mathematical Structures for Computer Science”, Sixth Edition, W.H.Freeman and Company, NY, 2007.
2. Trembly J.P and Manohar R, “Discrete Mathematical Structures with Applications to Computer Science”, Tata McGraw–Hill Pub. Co. Ltd, New Delhi, 2003.
3. Ralph. P. Grimaldi, “Discrete and Combinatorial Mathematics: An Applied Introduction”, Fourth Edition, Pearson Education Asia, Delhi, 2002.
4. Bernard Kolman, Robert C. Busby, Sharan Cutler Ross, “Discrete Mathematical Structures”, Fourth Indian reprint, Pearson Education Pvt Ltd., New Delhi, 2003.
5. Kenneth H.Rosen, “Discrete Mathematics and its Applications”, Fifth Edition, Tata McGraw – Hill Pub. Co. Ltd., New Delhi, 2003.
6. Richard Johnsonbaugh, “Discrete Mathematics”, Fifth Edition, Pearson Education Asia, New Delhi, 2002.



Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE : Data Structures
COURSE CODE : MSCS-2
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the data structures, the student shall be able to:

- To assess how the choice of data structures and algorithm design methods impacts the performance of programs.
 - To solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, heaps, binary search trees, and graphs and writing programs for these solutions.
-

COURSE OUTCOMES

After completion of the data structures, the student will be able to:

- To introduce the concept of data structures through ADT including List, Stack, Queues.
- To design and implement various data structure algorithms.
- To introduce various techniques for representation of the data in the real world.
- To develop application using data structure algorithms.

Block -1: INTRODUCTION

Definition of Algorithm and Data structure- Types of Data structure (linear and non- linear data structure) Linear data structure: Array, Stack, Queue, Linked List-doubly linked list and circular list - Recursive and non recursive algorithm.

Block – 2: NON-LINEAR DATA STRUCTURE

Binary Tree – notations, terminology, Representation, Binary tree Traversal and Application. Graph- Notations, Terminology-Representation, Traversal and Application. Performance analysis of an algorithm.-Tabular method.

Block -3: HEAP STRUCTURES

Min/Max heaps – Deaps – Leftist Heaps – Binomial Heaps – Fibonacci Heaps – Skew Heaps – Lazy-Binomial Heaps.

Block 4: SEARCH STRUCTURES

Binary Search Trees – AVL Trees – Red-Black trees – Multi-way Search Trees –B-Trees – Splay Trees – Tries.

Block 5: MULTIMEDIA STRUCTURES

Segment Trees – k-d Trees – Point Quad Trees – MX-Quad Trees – R-Trees – TVTrees

Reference books:

1. E. Horowitz, S.Sahni and Dinesh Mehta, Fundamentals of Data structures in C++, University Press, 2007.
2. E. Horowitz, S. Sahni and S. Rajasekaran, Computer Algorithms/C++, Second Edition, University Press, 2007.
3. G. Brassard and P. Bratley, Algorithmics: Theory and Practice, Prentice –Hall, 1988.
4. V.S. Subramanian, Principles of Multimedia Database systems, Morgan Kaufman, 1998.



Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE : Distributed Operating System
COURSE CODE : MSCS-13
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the **Distributed Operating System**, the student shall be able to:

- To study Distributed operating system concepts
 - To understand hardware, software and communication in distributed OS
 - To learn the distributed resource management components.
 - Practices to learn concepts of OS and Program the principles of Operating Systems
-

COURSE OUTCOMES

After completion of the **Distributed Operating System**, the student will be able to:

- Clear understanding on several resource management techniques like distributed shared memory and other resources
- Knowledge on mutual exclusion and Deadlock detection of Distributed operating system.
- Able to design and implement algorithms of distributed shared memory and commit protocols
- Able to design and implement fault tolerant distributed systems.

Block 1

Introduction – Operating System Definition – Functions of Operating System – Types of Advanced Operating System – Design Approaches – Synchronization Mechanisms – concepts of a Process – Critical Section Problem – Process Deadlock – Models of Deadlock – Conditions for Deadlock – System with single-Block 1 requests, Consumable Resources , Reusable Resources.

Block 2

Distributed Operating Systems: Introduction- Issues – Communication Primitives – Inherent Limitations –Lamport’s Logical Clock, Vector Clock, Global State, Cuts – Termination Detection – Distributed Mutual Exclusion – Non Token Based Algorithms – Lamport’s Algorithm - Token Based Algorithms –Distributed Deadlock Detection – Distributed Deadlock Detection Algorithms – Agreement Protocols

Block 3

Distributed Resource Management – Distributed File Systems – Architecture – Mechanisms – Design Issues – Distributed shared Memory – Architecture – Algorithm – Protocols – Design Issues – Distributed Scheduling – Issues – Components – Algorithms.

Block 4

Failure Recovery and Fault Tolerance – Concepts – Failure Classifications – Approaches to Recovery – Recovery in Concurrent Systems – Synchronous and Asynchronous Check pointing and Recovery –Check pointing in Distributed Database Systems – Fault Tolerance

Issues – Two-Phase and Nonblocking Commit Protocols – Voting Protocols – Dynamic Voting Protocols.

Block 5

Multiprocessor and Database Operating Systems –Structures – Design Issues – Threads – Process Synchronization – Processor Scheduling – Memory management – Reliability/Fault Tolerance – Database Operating Systems – concepts – Features of Android OS, Ubuntu, Google Chrome OS and Linux operating systems.

Reference Books:

1. Mukesh Singhal N.G.Shivaratri, “Advanced Concepts in Operating Systems”, McGraw Hill 2000.
2. Abraham Silberschatz, Peter B.Galvin, G.Gagne, “Operating Concepts”, 6th Edition Addison Wesley publications 2003.
3. Andrew S.Tanenbaum, Herbert bos “Modern Operating Systems”, 4th Edition, Pearson. 2014



Tamil Nadu Open University
School of Computer Science
Chennai – 15

Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE : Advanced Database Management Systems
COURSE CODE : MSCS-4
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the **Advanced Database Management Systems**, the student shall be able to:

- Model an application's data requirements using Entity Relationship Model and design Relational Database Schema based on the Entity Relationship Model and understand the need for Database Normalization.
 - Develop SQL Queries to retrieve data.
 - Creating XML Databases.
 - Design and Implement Distributed Databases from an application's data requirements.
 - Gain Knowledge on the Need for Database Security.
-

COURSE OUTCOMES

After completion of the **Advanced Database Management Systems**, the student will be able to:

- Know about the Various Datamodels and Works on Database Architecture
 - Knowledge patterns, Object Oriented Databases are well equipped.
-

Block 1: Relational Data Model, Relational Algebra and SQL

Relational Model Concepts – Super Key – Candidate Key – Primary Key – Alternate Key – Foreign Key – Relation Model Constraints and Relational Database Schemas – Update Operations – Transactions and Dealing with Constraint Violations – Unary and Binary Relational Operations in Relational Algebra – Queries in Relational Algebra – Basic SQL – SQL Data Definition and Data Types – Specifying Constraints in SQL – Basic Retrieval Queries in SQL – Insert, Delete and Update Statements in SQL – Views in SQL – Nested Queries – Triggers.

Block 2: Relational Database Design

Entity Relationship Model – Enhanced Entity Relationship Model – Mapping Entity Relationship Model to Relations – Mapping Enhance Entity Relationship Model to Relations – Database Normalization – Functional Dependencies – First Normal Form – Second Normal Form – Third Normal Form – Boyce-Codd Normal Form – Fourth Normal Form and Fifth Normal Form.

Block 3: XML: Extensible Markup Language

Structured, Semistructured, and Unstructured Data – XML Hierarchical Data Model – XML Documents, DTD, and XML Schema – Storing and Extracting XML Documents from Databases – XML Languages – Extracting XML Documents from Relational Databases – SQL Functions for Creating XML Data.

Block 4: Distributed Databases

Distributed Database Concepts – Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design – Overview of Concurrency Control and Recovery in Distributed Databases – Overview of Transaction Management in Distributed Databases – Query Processing and Optimization in Distributed Databases – Distributed Database Architectures – Distributed Catalog Management.

Block 5: Database Security

Database Security Issues – Discretionary Access Control Based on Granting and Revoking Privileges – Mandatory Access Control and Role-Based Access Control for Multilevel Security – SQL Injection – Statistical Database security – Flow Control – Encryption and Public Key Infrastructures – Privacy Issues and Preservation – Challenges to Maintaining Database Security.

Reference book:

1. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Seventh Edition, Pearson Education, 2017.
2. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Sixth Edition, Tata McGraw Hill, 2014.
3. C. J. Date, A. Kannan, S. Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.
4. Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems”, Fourth Edition, Tata McGraw Hill, 2010.



**Tamil Nadu Open University
School of Computer Science
Chennai – 15**

Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE : Object Oriented Programming Using C++

COURSE CODE : MSCS-5

COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Object Oriented Programming Using C++ , the student shall be able to:

- Understand the basic concepts of the Object-Oriented Programming Paradigm.
 - Gain Knowledge on Functions in C++ and the different types of Constructors in C++.
 - Understand Operator Overloading and the different types of Inheritance.
 - Gain Knowledge on Stream Classes, Files and Templates.
 - Understand the use of Exception Handling and Standard Template Library.
-

COURSE OUTCOMES

After completion of the Object Oriented Programming Using C++, the student will be able to:

- Describe the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects.
- Understand dynamic memory management techniques
- using pointers, constructors, destructors, etc
- Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.
- Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.
- Demonstrate the use of various OOPs concepts with the help of program

Block 1

Object-Oriented Programming (OOP) Paradigm – Basic Concepts of OOP – Benefits of OOP – Tokens – Keywords – Identifiers and Constants – Basic Data Types – User-Defined Data Types – Storage Classes – Derived Data Types – Symbolic Constants – Type Compatibility – Declaration of Variables – Dynamic Initialization of Variables – Reference Variables – Operators in C++ – Scope Resolution Operator – Member Dereferencing Operators – Memory Management Operators – Manipulators – Type Cast Operator – Expressions and Their Types – Special Assignment Expressions – Implicit Conversions – Operator Overloading – Operator Precedence – Control Structures.

Block 2

Functions in C++ – The Main Function – Function Prototyping – Call By Reference – Return by Reference – Inline Functions – Default Arguments – ‘const’ Arguments – Recursion – Function Overloading – Friend and Virtual functions – Math Library Functions – Classes and Objects – Specifying a Class – Defining Member Functions – Making an Outside Function Inline – Nesting of Member Functions – Private Member Functions – Arrays within a Class – Memory Allocation for Objects – Static Data Members – Static Member Functions – Array of Objects – Objects as Function Arguments – Friendly Functions – Returning Objects – ‘const’ Member Functions – Pointers to Members – Local Classes – Constructors – Parameterized Constructors – Multiple Constructors in a Class – Constructors with Default Arguments – Dynamic Initialization of Objects – Copy Constructors – Dynamic Constructors – Constructing Two-Dimensional Arrays – ‘const’ Objects – Destructors.

Block 3

Operator Overloading – Rules for Overloading Operators – Overloading Unary Operators – Overloading Binary Operators – Overloading Binary Operators Using Friends – Manipulation of Strings Using Operators – Type Conversions – Inheritance – Single Inheritance – Making a Private Member Inheritable – Multilevel Inheritance – Multiple Inheritance – Hierarchical Inheritance – Hybrid Inheritance – Virtual Base Classes – Abstract Classes – Constructors in Derived Class – Nesting of Classes – Pointers – Pointers to Objects – ‘this’ Pointer – Pointers to Derived Classes – Virtual Functions – Pure Virtual Functions – Virtual Constructors and Destructors – C++ Streams – C++ Stream Classes – Unformatted I/O Operations – Formatted Console I/O operations – Managing Output with Manipulators.

Block 4

Files – Classes for File Stream Operations – Opening and Closing a File – Detecting End-of-File – Open () File Modes – File Pointers and their Manipulators – Sequential Files – Random Access Files – Error Handling during File Operations – Command-Line Arguments – Templates – Class Templates – Class Templates with Multiple Parameters – Function Templates – Function Templates with Multiple Parameters – Overloading of Template Functions – Member Function Templates – Non-Type Template Arguments.

Block 5

Exception Handling –Exception Handling Mechanism –Throwing Mechanism –Catching Mechanism –Rethrowing an Exception –Specifying Exceptions –Exceptions in Constructors and Destructors –Exceptions in Operator Overloaded Functions – Standard Template Library

(STL) –Components of STL –Containers –Algorithms –Iterators –Application of Container Classes –Function Objects– Creating String Objects – Manipulating String Objects – Relational Operations – Accessing Characters in Strings – Comparing and Swapping.

Reference Books:

1. E Balagurusamy, “Object Oriented Programming with C++”, Sixth Edition, McGraw Hill Education (India) Private Limited, 2013.
2. Herbert Schildt, “C++: The Complete Reference”, Fifth Edition, McGraw-Hill Education, 2012.



Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE : Data Structure Using C++ Laboratory
COURSE CODE : MSCS-P1
COURSE CREDIT : 02

COURSE OBJECTIVES

While studying the Data Structure Using C++ Laboratory, the student shall be able to:

- develop skills to design and analyze simple linear and non linear data structures.
 - strengthen the ability to the students to identify and apply the suitable data structure for the given real world problem.
 - Enable them to gain knowledge in practical applications of data structures
-

COURSE OUTCOMES

After completion of the Data Structure Using C++ Laboratory, the student will be able to:

- Understand basic data structures such as arrays, linked lists, stacks and queues.
- Solve problem involving graphs, trees and heaps.
- Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data.

Exercises:

1. C++ programs to implement the following using an array.
a) Stack ADT b) Queue ADT
2. C++ programs to implement the following using a singly linked list.
a) Stack ADT b) Queue ADT
3. C++ program to implement the deque (double ended queue) ADT using a doubly linked list.
4. C++ program to perform the following operations:
 Insert an element into a binary search tree.
 Delete an element from a binary search tree.
 Search for a key element in a binary search tree.
5. C++ program to implement circular queue ADT using an array.
6. C++ programs that use non-recursive functions to traverse the given binary tree in
a) Preorder b) inorder c) postorder.
7. C++ programs for the implementation of bfs and dfs for a given graph.
8. C++ programs for implementing the following sorting methods:
a) Quick sort b) Merge sort c) Heap sort
9. C++ program to perform the following operations
a) Insertion into a B-tree b) Deletion from a B-tree
10. C++ program to perform the following operations
a) Insertion into an AVL-tree b) Deletion from an AVL-tree



Tamil Nadu Open University
School of Computer Science
Chennai – 15

Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE	:	Advanced Database Management Systems Laboratory
COURSE CODE	:	MSCS-P2
COURSE CREDIT	:	02

COURSE OBJECTIVES

While studying the Advanced Database Management Systems Laboratory, the student shall be able to:

- To explain basic database concepts, applications, data models, schemas and instances.
 - To demonstrate the use of constraints and relational algebra operations.
 - Describe the basics of SQL and construct queries using SQL.
-

COURSE OUTCOMES

After completion of the Advanced Database Management Systems Laboratory, the student will be able to:

- To create employee profile management
- To develop payroll processing and database oriented financial operations
- To work with the various databases

Exercises:

1. Display all the employees in alphabetical order from employee table.
2. Change the basic salary Rs,3000 where basic salary less than 2500 from employee table.
3. Change the basic_sal = 3000 where job in clerk from employee table.
4. Delete all records from Dept table.
5. Add a column “ Telephone_No“ of data type ‘number’ and size = ‘10’ to the employee table.
6. Create an index for client _no on client_mast table.
7. Display the length of each employee name from employee table.
8. Find average salary per job in each department _no.
9. Display only those jobs where max sal >=3000.
10. Find out the difference between highest and lowest salaries.
11. Display the sequence root of 81.
12. Display the total number of working days of each employee from employee table.
13. Count the number of products having price greater than equal to 1200.
14. Calculate the average price of all the products from product –mast table.
15. Find out the employees who earn the highest salary in each dept_no
16. Display all clerks from emp table using cursor.
17. Update all Sal < 1000 to Sal = 2000 using cursor.
18. Create a function for Simple Interest.
19. Create a recursive function for Fibonnaci series.
20. Create a package for checking the Even or Odd numbers.
21. Create a package for bank transactions for account debit and credit.



Mater of Science in Computer Science - Syllabus – II Semester (Distance Mode)

COURSE TITLE : Analysis of Algorithms
COURSE CODE : MSCS-6
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Analysis of Algorithms, the student shall be able to:

- To learn effective problem solving in Computing applications and analyze the algorithmic procedure to determine the computational complexity of algorithms.
-

COURSE OUTCOMES

After completion of the Analysis of Algorithms, the student will be able to:

- It gives stepwise procedure to solve problems.
- The Problems can be broken down into small pieces for program development.
- Know Efficient approach of solving problems by a model of computations

Block 1: Introduction: Algorithm Definition – Algorithm Specification – Performance Analysis-Asymptotic Notations. Elementary Data Structures: Stacks and Queues – Trees – Dictionaries – Priority Queues – Sets and Disjoint Set Union – Graphs

Block 2: Divide and Conquer: The General Method – Defective Chessboard – Binary Search – Finding The Maximum And Minimum – Merge Sort – Quick Sort – Selection – Strassen’s Matrix Multiplication.

Block 3: The Greedy Method:

General Method - Container Loading - Knapsack Problem - Tree Vertex Splitting – Job Sequencing With Deadlines - Minimum Cost Spanning Trees - Optimal Storage On Tapes – Optimal Merge Patterns - Single Source Shortest Paths.

Block 4: Dynamic Programming:

The General Method – Multistage Graphs – All-Pairs Shortest Paths – Single-Source Shortest Paths - Optimal Binary Search Trees - String Editing - 0/1 Knapsack - Reliability Design - The Traveling Salesperson Problem - Flow Shop Scheduling. Basic Traversal and Search Techniques: Techniques for Binary Trees – Techniques for Graphs – Connected Components and Spanning Trees – Biconnected Components and DFS.

Block 5: Backtracking:

The General Method – The 8-Queens Problem – Sum of Subsets – Graph Coloring – Hamiltonian Cycles – Knapsack Problem Branch and Bound: Least Cost searchhod - 0/1 Knapsack Problem.

Reference books:

1. Ellis Horowitz, Satraj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms, Universities Press, Second Edition, Reprint 2009.
2. Data Structures Using C - Langsam, Augenstein, Tenenbaum, PHI
3. Data structures and Algorithms, V.Aho, Hopcroft, Ullman , LPE
4. Introduction to design and Analysis of Algorithms - S.E. Goodman, ST. Hedetniem-TMH.
5. Carlos A.Coello Coello, Gary B.Lamont, David A.Van Veldhuizen, “Evolutionary Algorithms for Solving Multi-Objective Problems”, Springer 2nd Edition, 2007.



Mater of Science in Computer Science - Syllabus – II Semester (Distance Mode)

COURSE TITLE : Cloud Computing
COURSE CODE : MSCS-7
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Cloud Computing, the student shall be able to:

- Provide students with the comprehensive and in-depth knowledge of Cloud Computing concepts, technologies, architecture and applications by introducing and researching state-of-the-art in Cloud Computing fundamental issues, technologies, applications and implementations.
 - To expose the students to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.
-

COURSE OUTCOMES

After completion of the Cloud Computing, the student will be able to:

- Provide you with a good understanding of cloud computing and a systematic knowledge of the fundamental technologies, architecture, and security.
-

Block 1: COMPUTING BASICS

Cloud computing definition- Characteristics- Benefit-Challenges- Distributed Systems- Virtualization-Service-oriented computing- Utility-oriented computing- Building Cloud Computing environments- computing platforms & technologies - Cloud Models – Cloud Service Examples - Cloud Based Services & Applications - Cloud concepts and Technologies.

Block 2: VIRTUALIZATION, CLOUD SERVICES AND PLATFORMS

Virtualization:Virtualization- Characteristics- taxonomy-types- Pros and Cons- Examples
Architecture: Reference model- types of clouds- Compute Service - Storage Services - Cloud Database Services - Application Services - Content Delivery Services - Analytics Services - Deployment And Management Service - Identity And Access Management Services - Open Source Private Cloud Software.

Block 3: CLOUD APPLICATION DESIGN AND DEVELOPMENT

Design consideration- Reference Architecture for Cloud Application - Cloud Application Design Methodologies - Data Storage Approaches- Development in Python: Design Approaches – Application: Image Processing - Document Storage - Map Reduce - Social Media Analytics.

Block 4: PYTHON FOR CLOUD

Introduction- Installing Python- Data types & Data Structures- Control Flow- Functions- Modules- Packages- File Handling-Date/Time Operations – Classes- Python for Cloud: Amazon Web Services –Google Cloud Platform - Windows Azure –Map Reduced –Packages of Interest – Designing a RESTful Web API.

Block 5: BIG DATA ANALYTICS, MULTIMEDIA CLOUD & CLOUD SECURITY

Big Data Analytics: Clustering Big data - Classification of Big Data – Recommendation systems. Multimedia Cloud: Case Study: Live Video Stream App - Streaming Protocols – Case Study: Video Transcoding App-Cloud Security: CSA Cloud Security Architecture - Authentication - Authorization - Identity and Access management - Data Security - Key Management- Auditing- Cloud for Industry, Healthcare & Education.

Reference book:

1. Buyya, Vecciola and Selvi, Mastering Cloud Computing: Foundations and Applications Programming, Tata McGraw Hill, 2013.
2. ArshdeepBahga, Vijay Madiseti, “Cloud Computing: A Hands – On Approach” Universities press (India) Pvt. limited 2016.
3. Rittinghouse and Ransome, Cloud Computing: Implementation, Management, and Security, CRC Press, 2016.
4. Michael Miller “Cloud Computing Web based application that change the way you work and collaborate online”. Pearson edition, 2008.
5. Kris Jamsa, Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More, Jones & Bartlett Learning, 2012.



Mater of Science in Computer Science - Syllabus – II Semester (Distance Mode)

COURSE TITLE : Internet of Things
COURSE CODE : MSCS-8
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Internet of Things, the student shall be able to:

- Gain knowledge on bases of Internet of Things (IoT), IoT Architecture, and the Protocols related to IoT; and understand the concept of the Web of Thing and the relationship between the IoT and WoT.
-

COURSE OUTCOMES

After completion of the Internet of Things, the student will be able to:

- Gain the basic knowledge about IoT and they will be able to use IoT related products in real life.
- It helps to rely less on physical resources and started to do their work smarter.

Block 1: INTRODUCTION To IoT:

Internet of Things - Physical Design- Logical Design- IoT Enabling Technologies - IoT Levels and Deployment Templates - Domain Specific IoTs - IoT and M2M - IoT System Management with NETCONF-YANG- IoT Platforms Design Methodology.

Block 2: IoT ARCHITECTURE:

M2M high-level ETSI architecture - IETF architecture for IoT - OGC architecture - IoT reference model - Domain model - information model - functional model - communication model - IoT reference architecture

Block 3: IoT PROTOCOLS:

Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFID Protocols – Unified Data Standards – Protocols – IEEE 802.15.4 – BACNet Protocol – Modbus– Zigbee Architecture – Network layer – 6LowPAN - CoAP - Security

Block 4: WEB OF THINGS:

Web of Things versus Internet of Things – Two Pillars of the Web – Architecture Standardization for WoT– Platform Middleware for WoT – Unified Multitier WoT Architecture – WoT Portals and Business Intelligence. Cloud of Things: Grid/SOA and Cloud Computing – Cloud Middleware – Cloud Standards – Cloud Providers and Systems – Mobile Cloud Computing – The Cloud of Things Architecture.

Block 5: APPLICATIONS:

The Role of the Internet of Things for Increased Autonomy and Agility in Collaborative Production Environments - Resource Management in the Internet of Things: Clustering, Synchronisation and Software Agents. Applications - Smart Grid – Electrical Vehicle Charging.

Reference books:

1. Arshdeep Bahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015.
2. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), "Architecting the Internet of Things", Springer, 2011.
3. Jan Höller, Vlasios Tsiatsis , Catherine Mulligan, Stamatis , Karnouskos, Stefan Avesand. David Boyle, "From Machine-to-Machine to the Internet of Things - Introduction to a New Age of Intelligence", Elsevier, 2014.
4. Networks, Crowds, and Markets: Reasoning About a Highly Connected World - David Easley and Jon Kleinberg, Cambridge University Press - 2010.
5. Olivier Hersent, David Boswarthick, Omar Elloumi , "The Internet of Things – Key applications and Protocols", Wiley, 2012.



Master of Science in Computer Science - Syllabus – II Semester (Distance Mode)

COURSE TITLE : Data Mining
COURSE CODE : MSCS-9
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Data Warehousing and Data Mining, the student shall be able to:

- Understand the need for Knowledge Discovery and Gain Insight on the Techniques for Data Preprocessing.
 - Gain Knowledge on Schemas of a Data Warehouse, OLAP and Cube Technology.
 - Understand the Methods for Mining Frequent Patterns, Associations and Correlations.
 - Gain Knowledge on the Basic Concepts and Methods for Classification.
 - Gain Knowledge on the Basic Concepts and Methods for Data Clustering.
-

COURSE OUTCOMES

After completion of the Data Warehousing and Data Mining, the student will be able to:

- Brief the introduction about the data warehousing their architecture.
- Get clear idea about the Knowledge discovery from Databases
- Get the concepts of Data mining functionalities and mining rules
- Describe the Classification and prediction and propagations
- Understand the Clustering analysis and their concepts

Block– 1

Introduction – Kinds of Data and Patterns That Can Be Mined – Technologies Used – Kinds of Applications Targeted – Major Issues in Data Mining – Data Objects and Attribute Types – Basic Statistical Description of Data – Data Visualization – Measuring Data Similarity and Dissimilarity – Data Preprocessing – Data Cleaning – Data Integration – Data Reduction – Data Transformation and Data Discretization.

Block– 2

Data warehouse – Basic Concepts – Operational Database Systems and Data Warehouses – Architecture of a Data Warehouse – Extract, Transform and Load – Metadata – Data Cube and OLAP – Multidimensional Data Model – Star Schema – Snowflake Schema – Starflake Schema – Fact Constellation Schema – Dimensions – Measures – OLAP Operations – Starlet Query Model for Querying Multidimensional Databases – Data Warehouse Design and Usage – Data Warehouse Implementation – ROLAP – MOLAP – HOLAP – Data Generalization by Attribute-Oriented Induction – Data Cube Technology – Data Cube Computation: Preliminary Concepts – Data Cube Computation Methods – Processing Queries by Exploring Cube Technology – Multidimensional Data Analysis in Cube Space.

Block– 3

Mining Frequent Patterns, Associations, and Correlations – Basic Concepts – Market Basket Analysis – Frequent Itemsets, Closed Itemsets, and Association Rules – Frequent Itemset Mining Methods – Apriori Algorithm – Finding Frequent Itemsets by Confined Candidate Generation – Generating Association Rules from Frequent Itemsets – Improving the Efficiency of Apriori – Pattern-Growth Approach for Mining Frequent Itemsets – Mining Frequent Itemsets Using Vertical Data – Mining Closed and Max Patterns – Pattern Evaluation Methods – Association Analysis to Correlation Analysis – Comparison of Pattern Evaluation Measures – Pattern Mining in Multilevel, Multidimensional Space – Mining Multilevel Associations – Mining Multidimensional Associations – Mining Quantitative Association Rules – Mining Rare Patterns and Negative Patterns – Constraint-Based Frequent Pattern Mining – Metarule-Guided Mining of Association Rules – Constraint-Based Pattern Generation: Pruning Pattern Space and Pruning Data Space.

Block– 4

Classification – General Approach to Classification – Decision Tree Induction – Attribute Selection Measures – Tree Pruning – Scalability and Decision Tree Induction – Visual Mining for Decision Tree Induction – Bayes Classification Methods – Bayes’ Theorem – Naïve Bayesian Classification – Rule-Based Classification – Using IF-THEN Rules for Classification – Rule Extraction from a Decision Tree – Rule Induction Using a Sequential Covering Algorithm – Model Evaluation and Selection – Metrics for Evaluating Classifier Performance – Holdout Method and Random Subsampling – Cross-Validation – Bootstrap – Model Selection Using Statistical Tests of Significance – Comparing Classifiers Based on Cost-Benefit and ROC Curves – Techniques to Improve Classification Accuracy – Ensemble Methods – Bagging – Boosting and AdaBoost – Random Forests – Improving Classification Accuracy of Class-Imbalanced Data – Classification by Backpropagation – Support Vector Machines.

Block– 5

Cluster Analysis – Requirements for Cluster Analysis – Overview of Basic Clustering Methods – Partitioning Methods – k -Means: A Centroid-Based Technique – k -Medoids: A Representative Object-Based Technique – Hierarchical Methods – Agglomerative versus Divisive Hierarchical Clustering – Distance Measures in Algorithmic Methods – BIRCH: Multiphase Hierarchical Clustering Using Clustering Feature Trees – Chameleon: Multiphase Hierarchical Clustering Using Dynamic Modeling – Probabilistic Hierarchical Clustering – Density-Based Methods – DBSCAN: Density-Based Clustering Based on Connected Regions with High Density – OPTICS: Ordering Points to Identify the Clustering Structure – DENCLUE: Clustering Based on Density Distribution Functions – Grid-Based Methods – STING: Statistical Information Grid – CLIQUE: An Apriori-like Subspace Clustering Method – Evaluation of Clustering – Assessing Clustering Tendency – Determining the Number of Clusters – Measuring Clustering Quality.

Reference Books:

1. Jiawei Han, Micheline Kamber, Jian Pei, “Data Mining: Concepts and Techniques”, Third Edition, Morgan Kaufmann Publishers, 2011.
2. Arun K Pujari, “Data Mining Techniques”, Fourth Edition, Orient BlackSwan, 2016.



Tamil Nadu Open University
School of Computer Science
Chennai – 15

Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE : Application Programming for Mobile Devices
COURSE CODE : MSCS-10
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Application Programming for Mobile Devices, the student shall be able to:

- Understand system requirements for mobile applications
 - Generate suitable design using specific mobile development frameworks
 - Generate mobile application design
 - Implement the design using specific mobile development frameworks
 - Deploy the mobile applications in marketplace for distribution
-

COURSE OUTCOMES

After completion of the Application Programming for Mobile Devices, the student will be able to:

- Describe the requirements for mobile applications
 - Explain the challenges in mobile application design and development
 - Develop design for mobile applications for specific requirements
 - Implement the design using Android SDK
 - Implement the design using Objective C and iOS
 - Deploy mobile applications in Android and iPhone marketplace for distribution
-

Block 1: INTRODUCTION

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

Block 2: BASIC DESIGN

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

Block 3: ADVANCED DESIGN

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

Block 4: TECHNOLOGY I - ANDROID

Introduction – Establishing the development environment – Android architecture – Activities and views – Interacting with UI – Persisting data using SQLite – Packaging and deployment – Interaction with server side applications – Using Google Maps, GPS and Wifi – Integration with social media applications.

Block 5: TECHNOLOGY II - IOS

Introduction to Objective C – iOS features – UI implementation – Touch frameworks – Data persistence using Core Data and SQLite – Location aware applications using Core Location and Map Kit – Integrating calendar and address book with social media application – Using Wifi - iPhone marketplace.

Reference books:

1. <http://developer.android.com/develop/index.html>
2. Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", Wrox, 2012
3. Charlie Collins, Michael Galpin and Matthias Kappler, "Android in Practice", DreamTech, 2012
4. James Dovey and Ash Furrow, "Beginning Objective C", Apress, 2012
5. David Mark, Jack Nutting, Jeff LaMarche and Frederic Olsson, "Beginning iOS 6 Development: Exploring the iOS SDK", Apress, 2013.



**Tamil Nadu Open University
School of Computer Science
Chennai – 15**

Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE	:	Application Programming Laboratory for Mobile Devices
COURSE CODE	:	MSCS-P3
COURSE CREDIT	:	02

COURSE OBJECTIVES

While studying the Application Programming Laboratory for Mobile Devices, the student shall be able to:

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

COURSE OUTCOMES

After completion of the Application Programming Laboratory for Mobile Devices, the student will be able to:

- Design and Implement various mobile applications using emulators.
- Deploy applications to hand-held devices.

Exercises:

1. Develop an application that uses GUI components, Font and Colours
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Create an application that draws basic graphical primitives on the screen.
5. Develop an application that makes use of database.
6. Develop an application that makes use of RSS Feed.
7. Implement an application that implements Multi threading
8. Develop a native application that uses GPS location information.
9. Implement an application that writes data to the SD card.
10. Implement an application that creates an alert upon receiving a message.
11. Create a mobile application that creates alarm clock



Tamil Nadu Open University
School of Computer Science
Chennai – 15

Mater of Science in Computer Science - Syllabus – II Semester (Distance Mode)

COURSE TITLE : Algorithm Laboratory
COURSE CODE : MSCS-P4
COURSE CREDIT : 02

COURSE OBJECTIVES

While studying the Algorithm Laboratory, the student shall be able to:

- To familiarize various data structure implementations.
 - To implement heap and various tree structures like AVL, Red-black, B-Tree and segment trees.
 - To understand efficient implementation of line segment intersection.
 - To understand various search structures.
 - To get understanding of problem to program mapping.
-

COURSE OUTCOMES

After completion of the Algorithm Laboratory, the student will be able to:

- Achieve programming skill to convert a problem to a programming logic.
- Apply suitable data structure for the problem in hand.
- Implement heap and various tree structures like AVL, Red-black, B-Tree and segment trees.
- Understand the usage of data structures for geometric problems.
- Understand the importance of height balancing in search structures.

Exercises:

1. Implement Insertion Sort (The program should report the number of comparisons)
2. Implement Merge Sort(The program should report the number of comparisons)
3. Implement Heap Sort (The program should report the number of comparisons)
4. Implement Randomized Quick sort (The program should report the number of comparisons)
5. Implement Radix Sort.
6. Implement Searching Techniques
7. Implementation of Recursive function.
8. Array and Linked list implementation of Stack and Queue.
9. Implementation of Single, Double and circular Linked List
10. Creation and traversal of Binary Search Tree.



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Web Technology

COURSE CODE : MSCS-11

COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Web Technology, the student shall be able to:

- Explore the backbone of web page creation by developing .NET skill.
- Enrich knowledge about HTML control and web control classes
- Provide depth knowledge about ADO.NET
- Understand the need of usability, evaluation methods for web services

COURSE OUTCOMES

After completion of the Web Technology, the student will be able to:

- Design a web page with Web form fundamentals and web control classes
- Recognize the importance of validation control, cookies and session
- Apply the knowledge of ASP.NET object, ADO.NET data access and SQL to develop a client server model.
- Recognize the difference between Data list and Data grid controls in accessing data.

Block 1: OVERVIEW OF ASP.NET

The .NET framework – Learning the .NET languages Data types – Declaring variables- Scope and Accessibility- Variable operations- Object Based manipulation- Conditional Structures- Loop Structures- Functions and Subroutines. Types, Objects and Namespaces : The Basics about Classes- Value types and Reference types- Advanced class programming- Understanding name spaces and assemblies. Setting Up ASP.NET and IIS.

Block 2: Developing ASP.NET Applications

ASP.NET Applications: ASP.NET applications– Code behind- The Global.asax application file- Understanding ASP.NET Classes- ASP.NET Configuration. Web Form fundamentals: A simple page applet- Improving the currency converter- HTML control classes- The page class- Accessing HTML server controls. Web controls: Web Control Classes – AutoPostBack and Web Control events- Accessing web controls. Using Visual Studio.NET: Starting a Visual Studio.NET Project- Web form Designer- Writing code- Visual studio.NET debugging. Validation and Rich Controls: Validation- A simple Validation example- Understanding regular expressions- A validated customer form. State management - Tracing, Logging, and Error Handling.

Block 3: Working with Data

Overview of ADO.NET - ADO.NET and data management- Characteristics of ADO.NET- ADO.NET object model. ADO.NET data access : SQL basics– Select , Update, Insert, Delete statements- Accessing data- Creating a connection- Using a command with a DataReader - Accessing Disconnected data - Selecting multiple tables – Updating Disconnected data. Data binding: Single value Data Binding- Repeated value data binding- Data binding with data bases. Data list – Data grid – Repeater – Files, Streams and Email – Using XML

Block 4:

Web Services - Web services Architecture: Internet programming then and now- WSDL– SOAP- Communicating with a web service-Web service discovery and UDDI. Creating Web services: Web service basics- The Stock Quote web service – Documenting the web service- Testing the web service- Web service Data types- ASP.NET intrinsic objects. Using web services: Consuming a web service- Using the proxy class- An example with TerraService.

Block 5:

Advanced ASP.NET - Component Based Programming: Creating a simple component – Properties and state- Database components- Using COM components. Custom controls: User Controls- Deriving Custom controls. Caching and Performance Tuning: Designing and scalability– Profiling- Catching- Output catching- Data catching. Implementing security: Determining security requirements- The ASP.NET security model- Forms authentication- Windows authentication.

Reference Books:

1. Mathew Mac Donald, “ASP.NET Complete Reference”, TMH 2005.
2. Crouch Matt J, “ASP.NET and VB.NET Web Programming”, Addison Wesley 2002.
3. J.Liberty, D.Hurwitz, “Programming ASP.NET”, Third Edition, O’REILLY, 2006.



Tamil Nadu Open University
School of Computer Science
Chennai – 15

Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Network Security
COURSE CODE : MSCS-12
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Network Security, the student shall be able to:

- To understand necessary Approaches and Techniques to build protection mechanisms in order to secure computer networks.
 - To know about the malicious software & firewalls.
-

COURSE OUTCOMES

After completion of the Network Security, the student will be able to:

- Understand the fundamentals of networks security, security architecture, threats and vulnerabilities
- Apply the various Authentication schemes to simulate different applications.
- Understand various Security practices and System security standards

Block 1: Introduction:

Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, A Model for Internetwork Security, Internet Standards and the Internet Society.

Block 2: Symmetric Encryption and Message Confidentiality: Symmetric Encryption Principles, Symmetric Block Encryption Algorithms, Stream Ciphers and RC4, Cipher Block Modes of Operation, Location of Encryption Devices, Key Distribution. **Public-key Cryptography and Message Authentication:** Approaches to Message Authentication, Secure Hash Functions and HMAC, Public-Key Cryptography Principles, Public-Key Cryptography Algorithms, Digital Signatures, Key Management.

Block 3: Authentication Applications:

Kerberos, x.509 Authentication Service, Public-Key Infrastructure. **Electronic Mail Security:** Pretty Good Privacy (PGP), S/MIME.

Block 4: IP Security:

IP Security Over view, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations. **Web Security:** Web Security Considerations, Secure Socket Layer(SSL) and Transport Layer Security(TLS), Secure Electronic Transaction(SET). **Network Management Security:** Basic Concepts of SNMP, SNMPv1 CommBlock 1y Facility, SNMPv3.

Block 5: Intruders:

Intruders, Intrusion Detection, Password Management. **Malicious Software:** Virus and Related Threats, Virus Countermeasures, Distributed Denial of Service Attacks. **Firewalls:** Firewall Design Principles, Trusted Systems, Common Criteria for Information Technology Security Evaluation.

Reference Books:

1. William Stallings, “Network Security Essentials Applications and Standards”Third Edition, Pearson Education, 2008.



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Big Data
COURSE CODE : MSCS-13
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Big Data, the student shall be able to:

- The course provides grounding in basic and advanced methods to big data technology and tools, including MapReduce and Hadoop and its ecosystem.
-

COURSE OUTCOMES

After completion of the Big Data, the student will be able to:

- Able to apply Hadoop ecosystem components.
- Able to participate data science and big data analytics projects

Block-1: INTRODUCTION TO BIG DATA

Introduction – distributed file system – Big Data and its importance, Four Vs, Drivers for Big data, Big data analytics, Big data applications. Algorithms using map reduce, Matrix-Vector Multiplication by Map Reduce.

Block-2: INTRODUCTION HADOOP

Big Data – Apache Hadoop & Hadoop EcoSystem – Moving Data in and out of Hadoop – Understanding inputs and outputs of MapReduce - Data Serialization.

Block-3: HADOOP ARCHITECTURE

Hadoop Architecture, Hadoop Storage: HDFS, Common Hadoop Shell commands , Anatomy of File Write and Read., NameNode, Secondary NameNode, and DataNode, Hadoop MapReduce paradigm, Map and Reduce tasks, Job, Task trackers - Cluster Setup – SSH & Hadoop Configuration – HDFS Administering –Monitoring & Maintenance.

Block-4: HADOOP ECOSYSTEM AND) YARN

Hadoop ecosystem components - Schedulers - Fair and Capacity, Hadoop 2.0 New Features NameNode High Availability, HDFS Federation, MRv2, YARN, Running MRv1 in YARN.

Block-5: HIVE AND HIVEQL, HBASE

Hive Architecture and Installation, Comparison with Traditional Database, HiveQL – Querying Data - Sorting And Aggregating, Map Reduce Scripts, Joins & Subqueries, HBase concepts Advanced Usage, Schema Design, Advance Indexing - PIG, Zookeeper - how it helps in monitoring a cluster, HBase uses Zookeeper and how to Build Applications with Zookeeper.

Reference Books:

1. Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, “Professional Hadoop Solutions”, Wiley, ISBN: 9788126551071, 2015.
2. Chris Eaton, Dirk deroos et al. , “Understanding Big data ”, McGraw Hill, 2012.
3. Tom White, “HADOOP: The definitive Guide” , O Reilly 2012.



**Tamil Nadu Open University
School of Computer Science
Chennai – 15**

Mater of Science in Computer Science - Syllabus – I Semester (Distance Mode)

COURSE TITLE : Web Technology Laboratory
COURSE CODE : MSCS-P5
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Web Technology Laboratory, the student shall be able to:

- To develop a web application using java technologies
- To create fully functional website/web applications.

COURSE OUTCOMES

After completion of the Web Technology Laboratory, the student will be able to:

- Understand, analyze and apply the role of languages like HTML, ASP and protocols in the workings of the web and web applications
- Analyze a web page and identify its elements and attributes.
- Create dynamic web pages using JavaScript and VBScript (client side programming).
- Create interactive web applications using ASP.NET.
- Create XML documents and XML Schema.
- Build and consume web services.

Exercises

1. Create a table to implement with rowspan and colspan.
2. Implement all types of list.
3. Create 4 frames in a page & display 4 images in those frames.
4. Create login page with validation.
5. Implement and load image in an html page.
6. Implement onload event in an html page.
7. Create a program to display the following feedback form.
8. Create a simple ASP.NET program to display some Web Controls:
9. Create a program to display “Welcome To Radiant” in the form when the “click” button is clicked. The form title must be ASP.NET.
10. Create a program that displays a button in green color and it should change into yellow when the mouse moves over it.



Tamil Nadu Open University
School of Computer Science
Chennai – 15

Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Mini Project (Using Open Source Software)
COURSE CODE : MSCS-P6
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Project (Using Open Source Software), the student shall be able to:

- The objective of this mini project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good professionals

COURSE OUTCOMES

After completion of the Project (Using Open Source Software), the student will be able to:

- Demonstrate a through and systematic understanding of project contents.
- Understand methodologies and professional way of documentation and communication.
- Know the key stages in development of the project.
- Extend or use the idea in mini project for major project.



**Tamil Nadu Open University
School of Computer Science
Chennai – 15**

Mater of Science in Computer Science - Syllabus – IV Semester (Distance Mode)

COURSE TITLE : Project
COURSE CODE : MSCS-P7
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Project, the student shall be able to:

- To solve problems of a non-routine nature.
- To offer students to demonstrate their competence in laboratory work.
- To provide a vehicle for integrating the knowledge gained in various subjects of the degree course.
- To allow the exercise of the undergraduates' personal qualities - viz. maturity, initiative and creative ability.
- To apply communication skills, both oral and written, to communicate results, concepts and ideas.

COURSE OUTCOMES

After completion of the Project, the student will be able to:

- to plan and implement an investigative or developmental project given general objectives and guidelines
- get In-depth skill to use some laboratory / workshop equipment to process and characterize materials
- to analyze data to produce useful information and to draw conclusions by systematic deduction
- to work and study independently
- to communicate results, concepts, analyses and ideas in written and oral form



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Artificial Intelligence (Elective I)
COURSE CODE : MSCS-14
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Artificial Intelligence, the student shall be able to:

- Study the concepts of Artificial Intelligence.
 - Learn the methods of solving problems using Artificial Intelligence.
 - Introduce the concepts of Expert Systems and machine learning
-

COURSE OUTCOMES

After completion of the Artificial Intelligence, the student will be able to:

- Identify problems that are amenable to solution by AI methods.
- Identify appropriate AI methods to solve a given problem.
- Formalise a given problem in the language/framework of different AI methods.
- Implement basic AI algorithms.
- Design and carry out an empirical evaluation of different algorithms on a problem formalisation, and state the conclusions that the evaluation supports.

Block 1: INTRODUCTION

Introduction to Artificial Intelligence – Intelligent Agents – Agents and Environments - Good behavior – The Nature of Environments – Structure of Agents - Problem Solving - Problem Solving Agents – Agent Architectures and Hierarchical Control - Agents - Agent Systems – Hierarchical Control - Embedded and Simulated Agents - Acting with Reasoning.

Block 2: SEARCHING TECHNIQUES

Searching For Solutions – Uniformed Search Strategies - Avoiding Repeated States – Searching with Partial Information - Informed Search and Exploration – Informed Search Strategies – Heuristic Function – Local Search Algorithms and Optimistic Problems – Local Search in Continuous Spaces – Online Search Agents and Unknown Environments – Constraint Satisfaction Problems (CSP) – Backtracking Search and Local Search for CSPs – Structure of Problems - Adversarial Search – Games – Optimal Decisions in Games – Alpha-Beta Pruning – Imperfect Real-Time Decisions – Games that include an element of chance.

Block 3: KNOWLEDGE AND REASONING

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

Block 4: LEARNING

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

Block 5: AI PLANNING AND APPLICATIONS

AI Planning – Planning with State - Space Search – Partial-Order Planning – Planning Graphs – Planning with Propositional Logic- Hierarchical Task Network Planning – Conditional Planning - All applications – Language Models - Information Retrieval – Information Extraction – Machine Translation – Machine Learning - Symbol-Based – Machine Learning: Connectionist – Machine Learning - Social and Emergent -Robots

Reference Books:

1. Stuart Russell, Peter Norvig, “Artificial Intelligence: A Modern Approach”, Third Edition, Pearson Education / Prentice Hall of India, 2010.
2. Elaine Rich and Kevin Knight, “Artificial Intelligence”, Third Edition, Tata McGraw-Hill, 2010.
3. Bratko I, “Prolog Programming for Artificial Intelligence”, Addison-Wesley Educational Publishers Inc; Fourth Edition, 2011.
4. David L. Poole, Alan K. Mackworth, “Artificial Intelligence: Foundations of Computational Agents”, Cambridge University Press, 2010.
5. Ethem Alpaydin, “Introduction to Machine Learning (Adaptive Computation and Machine Learning series)”, The MIT Press; Second edition, 2009.
6. Patrick H. Winston. "Artificial Intelligence", Third edition, Pearson Edition, 2006.
7. Dan W.Patterson, “Introduction to Artificial Intelligence and Expert Systems”, PHI, 2006.
8. Nils J. Nilsson, “Artificial Intelligence: A new Synthesis”, Harcourt Asia Pvt. Ltd., 2000.



Tamil Nadu Open University
School of Computer Science
Chennai – 15

Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Mobile Computing (Elective I)
COURSE CODE : MSCS-15
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Mobile Computing, the student shall be able to:

- Understand the basic concepts of mobile
 - Be familiar with GPRS Technology
 - system Be exposed to Ad-Hoc networks
 - Gain knowledge about different mobile platforms and application development
-

COURSE OUTCOMES

After completion of the Mobile Computing, the student will be able to:

- Able to explain the basics of mobile system
- Able to develop mobile application
- Understand the Mobile Ad hoc networks and its routing
- Understand the different types of security features

Block -1

Mobile computing: Components of wireless environment- Challenges in Mobile environment- Mobile devices - Middleware and gateways - Wireless Internet - Smart clients - Three-tier Architecture- Design considerations for mobile computing— Mobility and Location based services – Active transactions - Device Technology – Device Connectivity – Voice technology – Personal digital assistant.

Block -2

Mobile computing through Internet- Mobile-enabled Applications - Developing Mobile GUIs – VUIs and Mobile Applications – Multichannel and Multi modal user interfaces – Synchronization and replication of Mobile Data - SMS architecture - Java card – GPRS – Mobile Computing through Telephony - Synchronization protocol - Context-aware applications.

Block -3

Mobile Communication: Wireless Transmission – Medium Access Control – Telecommunication Systems – Satellite Systems – Broadcast system – Wireless LAN – Mobile IP – Mobile TCP.

Block -4

ADHOC Wireless Network: Ad Hoc Wireless Network –MAC protocol – Routing protocols - Transport Layer Protocol - QOS – Energy Management.

Block -5

Wireless Sensor Network: Architecture and Design – Medium Access Control – Routing – Transport Layer – Energy model.

Reference Books:

1. Jochen Schiller, Mobile Communications, Second Edition, Revised 2008.
2. William Stallings, "Wireless Communications & Networks", Pearson Education, 2005.
3. C.Siva Ram Murthy, B.S. Manoj, "Ad Hoc Wireless Networks – Architectures and Protocols", 2nd Edition, Pearson Education. 2004
4. Ashok K Talukder, Roopa R Yavagal, "Mobile Computing", Tata McGraw Hill, 2005.
5. Jochen Burkhardt Dr.Horst Henn, Klaus Rintdoff,Thomas Schack, "Pervasive Computing", Pearson, 2009.
6. Fei Hu , Xiaojun Cao, " Wireless Sensor Networks Principles and Practice " CRC Press, 2010.



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : User Interface Design (Elective I)
COURSE CODE : MSCS-16
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the User Interface Design, the student shall be able to:

- Describe several fundamental user interface design principles.
- Explain the process of user interface design.
- Discuss how to design the user interface structure.
- Explain how to design the user interface standards.
- Be able to design a user interface.

COURSE OUTCOMES

After completion of the User Interface Design, the student will be able to:

- get understand the basics concepts of User interface design.
- describe the functions of Human Computer Interaction
- get clear idea about the Windows, Multimedia and Windows Layout– Test

Block 1: INTRODUCTION:

Human–Computer Interface – Characteristics Of Graphics Interface –Direct Manipulation Graphical System – Web User Interface –Popularity –Characteristic & Principles.

Block 2: HUMAN COMPUTER INTERACTION:

User Interface Design Process – Obstacles –Usability –Human Characteristics In Design – Human Interaction Speed –Business Functions –Requirement Analysis – Direct – Indirect Methods – Basic Business Functions – Design Standards – System Timings – Human Consideration In Screen Design – Structures Of Menus – Functions Of Menus– Contents Of Menu– Formatting – Phrasing The Menu – Selecting Menu Choice– Navigating Menus– Graphical Menus.

Block 3: WINDOWS:

Characteristics– Components– Presentation Styles– Types– Managements–Organizations– Operations– Web Systems– Device– Based Controls Characteristics–Screen – Based Controls – Operate Control – Text Boxes– Selection Control–Combination Control– Custom Control– Presentation Control.

Block 4: MULTIMEDIA:

Text For Web Pages – Effective Feedback– Guidance & Assistance– Internationalization– Accessibility– Icons– Image– Multimedia – Coloring.

Block 5: WINDOWS LAYOUT– TEST:

Prototypes – Kinds Of Tests – Retest – Information Search – Visualization – Hypermedia – WWW– Software Tools.

Reference Books:

1. Wilbent. O. Galitz ,“The Essential Guide To User Interface Design”, John Wiley& Sons, 2001.
2. Ben Sheiderman, “Design The User Interface”, Pearson Education, 1998.
3. Alan Cooper, “The Essential Of User Interface Design”, Wiley – Dream Tech Ltd., 2002.
4. click here to read more: <http://www.annaunivedu.in/2012/08/it2024-user-interface-design-syllabus.html#ixzz6PzXQwKei> more question bank and previous year question paper



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Unix Internals (Elective I)
COURSE CODE : MSCS-17
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Unix Internals, the student shall be able to:

- To understand the design of the UNIX operating system.
 - To become familiar with the various data structures used.
 - To learn the various low-level algorithms used in UNIX.
-

COURSE OUTCOMES

After completion of the Unix Internals, the student will be able to:

- To understand the design of the UNIX operating system.
 - To become familiar with the various data structures used.
 - To learn the various low-level algorithms used in UNIX.
-

Block 1: OVERVIEW

General Overview of the System: History –System structure –User perspective –Operating System Services –Assumptions about Hardware. Introduction to the Kernel Architecture of the UNIX Operating System –Introduction to System Concept -The Buffer Cache -Buffer headers –Structure of the Buffer Pool –Scenarios for Retrieval of a Buffer–Reading and Writing Disk Blocks –Advantages and Disadvantages of the Buffer Cache.

Block 2: FILE SUBSYSTEM

Internal Representation of Files: Inodes –Structure of a Regular File –Directories – Conversion of a Path Name to an Inode –Super Block –Inode Assignment to a New File – Allocation of Disk Blocks.

Block 3: SYSTEM CALLS FOR THE FILE SYSTEM

Open –Read –Write –File And Record Locking –Adjusting the Position of File I/O –lseek – close –File Creation –Creation of Special Files –Changing Directory –Root –Owner -Mode – stat And fstat –Pipes –dup –Mounting And Unmounting File Systems –link –unlink.

Block 4: PROCESSES

Process States and Transitions –Layout of System Memory –The Context of a Process – Saving the Context of a Process –Manipulation of the Process Address Space –Sleep -Process Control -Process Creation –Signals –Process Termination –Awaiting Process Termination – Invoking other Programs –User Id of a Process –Changing the size of a Process –Shell – System Boot and the INIT Process–Process Scheduling.

Block 5: MEMORY MANAGEMENT AND I/O

Memory Management Policies –Paging and Segmentation -Swapping –Demand Paging -The I/O Subsystem: Driver Interface –Disk Drivers –Terminal Drivers.

Reference book:

1. Maurice J. Bach, “The Design of the Unix Operating System”, First Edition, Pearson Education, 1999.
2. B. Goodheart, J. Cox, “The Magic Garden Explained”, Prentice Hall of India, 1986.
3. S. J. Leffler, M. K. McKusick, M. J. Karels and J. S. Quarterman., “The Design and Implementation of the 4.3 BSD Unix Operating System”, Addison Wesley, 1998.



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Big Data Query Languages (Elective I)
COURSE CODE : MSCS-18
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Big Data Query Languages, the student shall be able to:

- To familiarize with R Programming.
 - To understand data analysis using R and HADOOP Integrated Programming Environment.
 - To Understand Analytics for Big data ‘at Rest’ and Real-Time Analytical Processing for Big data ‘in Motion’.
 - To understand the Pig Data model and Pig scripts.
 - To learn way of Querying Big Data using Hive.
-

COURSE OUTCOMES

After completion of the Big Data Query Languages, the student will be able to:

- Design applications using R, HADOOP.
 - Design applications using RHADOOP& RHIPE.
 - Develop analytic applications for data Streams.
 - Develop Pig scripts for Big data applications.
 - Design Big data applications schema and use HIVE QL.
-

Block 1: INTRODUCTION TO R PROGRAMMING

Introduction to R –Vectors –Filtering –Matrices –Creating Matrices –Applying Functions to Matrix Rows and Columns –Lists –Creating List –General List Operations –Data Frames –Creating Data Frames –Matrix like Operations in Frames –Applying Functions to Data Frames –Factors and Tables –Math and Simulations in R –Input/Output –Reading and Writing Files –Graphics –Creating Three-Dimensional Plots –Linear Models –Non-linear models –Clustering.

Block 2: DATA ANALYSIS USING R AND HADOOP

Features of R Language -HADOOP Features –HDFS and MapReduce architecture –R and Hadoop Integrated Programming Environment-RHIPE Introduction –Architecture of RHIPE –RHIPE function reference. RHADOOP Introduction –Architecture of RHADOOP –RHADOOPfunction reference, SQL on HADOOP.

Block 3: ANALYTICS FOR BIG DATA STREAMS

IBM Pure Data Systems –Netezza’s Design Principles –The Netezza Appliance –Extending theNetezza Analytics –Real-Time Analytical Processing –InfoSphere Streams Basics –InfoSphereStreams Working –enterprise class –industry use cases –Indexing Data from Multiple Sources.

Block 4: PROGRAMMING WITH PIG

Introduction –installation and execution –PIG Data Model –PIG Latin –Input, Output-Relational Operators –User Defined Functions –Join Implementations –Integrating Pig with Legacy Code and Map Reduce –Developing and Testing Pig Latin Scripts –Embedding Pig Latin in Python –Evaluation Function in Java-Load Functions –Store Functions.

Block 5: PROGRAMMING WITH HIVE

Introduction –Data Types and File Formats –Databases in Hive –HiveQL: Data Definition –Data Manipulation –Queries –Views –Indexes –Schema Design.

Reference book:

1. Michael Berthold, David J. Hand, “Intelligent Data Analysis”, Springer, 2007.
2. Tom White “ Hadoop: The Definitive Guide” Third Edition, O’reilly Media, 2011.
3. Zikopoulos, P., Parasuraman, K., Deutsch, T., Giles, J., & Corrigan, D.V Harness the Power of Big Data The IBM Big Data Platform. McGraw Hill Professional, 2012.
4. Prajapati, V, “Big Data Analytics with R and Hadoop”, Packt Publishing Ltd, 2013.
5. Gates, A. Programming Pig.”O’Reilly Media, Inc.”, 2011.
6. Capriolo, E., Wampler, D., & Rutherglen, J., “Programming Hive”, O’ReillyMedia, Inc.”,2012
7. Norman Matloff, “The Art of R Programming: A Tour of Statistical Software Design”, NoStarch Press, 2011.
8. Jared P. Lander, “R for Everyone: Advanced Analytics and Graphics”, Addison-Wesley Data & Analytics Series, 2013



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Software Testing (Elective II)
COURSE CODE : MSCS-19
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Software Testing, the student shall be able to:

- To study various Software techniques
 - To study fundamental concepts in software testing
-

COURSE OUTCOMES

After completion of the Software Testing, the student will be able to:

- List a range of different software testing techniques and strategies and be able to apply specific(automated) Block 1 testing method to the projects.
- Distinguish characteristics of structural testing methods.
- Demonstrate the integration testing which aims to uncover interaction and compatibility problems as early as possible.
- Discuss about the functional and system testing methods.CO5) Demonstrate various issues for object oriented testing

Block 1:

Software Quality Assurance: software challenge - Software Quality – Software Quality factors – Software Quality Models- Software quality measurement and metrics – Software Quality Architecture.

Block 2:

Introduction to Software Testing: Overview- Purpose of testing – Objectives – Inspection and Testing – Testing and debugging – Debugging process – Software testing life cycle – Responsibility of test team leader.

Block 3:

Testing techniques: The V-Model – Testing techniques: Functional testing techniques – Non-functional testing techniques- Test metrics- Risk based testing – Extreme testing.

Block 4:

Automated testing: Introduction – process - Types of automated test – Code auditing – Coverage Monitoring – functional test – Load test – Test Management - Advantages and Disadvantages of Automated test - Alpha and Beta site testing programs.

Block 5:

Test Maturity Model: Human Issues and Challenges in testing.

Reference Books:

1. Software Quality Assurance, Nina S Godbole, Narosa Publishing House, 2008.
2. Software Quality Assurance, From Theory to Implementation, Daniel Galin, Pearson Education, 2004.
3. Software Quality Complete and Practices, R A Khan, K. Mustafa, SI Ahson , Narosa Publishing House, 2008.
4. Software Testing principles and Practices, Srinivasan Desikan, Gopalswamy Ramesh, Pearson Education, 2006.



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Supply Chain Management (Elective II)
COURSE CODE : MSCS-20
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Supply Chain Management, the student shall be able to:

- To help understand the importance of and major decisions in supply chain management for gaining competitive advantage.
-

COURSE OUTCOMES

After completion of the Supply Chain Management, the student will be able to:

- Ability to build and manage a competitive supply chain using strategies, models, techniques and information technology.
-

Block 1: INTRODUCTION

Supply Chain –Fundamentals –Evolution-Role in Economy -Importance -Decision Phases - Supplier-Manufacturer-Customer chain. -Enablers/ Drivers of Supply Chain Performance. Supply chain strategy -Supply Chain Performance Measures.

Block 21: STRATEGIC SOURCING

Outsourcing –Make Vs buy -Identifying core processes -Market Vs Hierarchy -Make Vs buy continuum -Sourcing strategy -Supplier Selection and Contract Negotiation. Creating a world class supply base-Supplier Development -World Wide Sourcing.

Block 3: SUPPLY CHAIN NETWORK

Distribution Network Design –Role -Factors Influencing Options, Value Addition – Distribution Strategies -Models for Facility Location and Capacity allocation. Distribution Center Location Models. Supply Chain Network optimization models. Impact of uncertainty on Network Design -Network Design decisions using Decision trees.

Block 4: PLANNING DEMAND, INVENTORY AND SUPPLY

Managing supply chain cycle inventory. Uncertainty in the supply chain –Analyzing impact of supply chain redesign on the inventory -Risk Pooling -Managing inventory for short life - cycle products -multiple item -multiple location inventory management. Pricing and Revenue Management

Block 5: CURRENT TRENDS

Supply Chain Integration -Building partnership and trust in SC Value of Information: BullwhipEffect -Effective forecasting -Coordinating the supply chain. . SC Restructuring -SC Mapping -SC process restructuring, Postpone the point of differentiation –IT in Supply Chain -Agile Supply Chains -Reverse Supply chain. Agro Supply Chains

Reference book:

1. Janat Shah, Supply Chain Management –Text and Cases, Pearson Education, 2009.
2. Sunil Chopra and PeterMeindl, Supply Chain Management-Strategy Planning and Operation, PHI Learning / Pearson Education, Sixth edition, 2015.
3. Ballou Ronald H, Business Logistics and Supply Chain Management, Pearson Education, 5thEdition, 2007.
4. David Simchi-Levi, Philip Kaminsky, Edith Simchi-Levi, Designing and Managing the Supply Chain: Concepts, Strategies, and Cases, Tata McGraw-Hill, 2005.
5. Altekar Rahul V, Supply Chain Management-Concept and Cases, PHI, 2005.
6. Shapiro Jeremy F, Modeling the Supply Chain, Cengage, Second Reprint , 2002.
7. Joel D. Wisner, G. Keong Leong, Keah-Choon Tan, Principles of Supply Chain Management-A Balanced Approach, South-Western, Cengage, 2012.



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Block chain Technologies (Elective II)
COURSE CODE : MSCS-21
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Block chain Technologies, the student shall be able to:

- To understand Blockchain's fundamental components, and examine decentralization using blockchain.
 - To explain how cryptocurrency works, from when a transaction is created to when it is considered part of the Blockchain.
 - To explain the components of Ethereum and Programming Languages for Ethereum.
 - To study the basics of Hyperledger and Web3.
 - To know about alternative Blockchains and Blockchain projects in different domains.
-

COURSE OUTCOMES

After completion of the Block chain Technologies, the student will be able to:

- Understand the technology components of Blockchain and how it works behind the scenes.
- Identify different approaches to developing decentralized applications.
- Understand Bitcoin and its limitations by comparing with other alternative coins.
- Devise solution using the Ethereum model.
- Understand and use Hyperledger and its development framework.
- Track alternative Blockchains and emerging trends in Blockchain

Block 1: INTRODUCTION TO BLOCKCHAIN

History of Blockchain –Types of Blockchain –Consensus –Decentralization using Blockchain –Blockchain and Full Ecosystem Decentralization –Platforms for Decentralization

Block 2: INTRODUCTION TO CRYPTOCURRENCY

Bitcoin –Digital Keys and Addresses –Transactions –Mining –Bitcoin Networks and Payments –Wallets –Alternative Coins –Theoretical Limitations –Bitcoin limitations –Name coin –Prime coin –Zcash –Smart Contracts –Ricardian Contracts.

Block 3: ETHEREUM

The Ethereum Network –Components of Ethereum Ecosystem –Ethereum Programming Languages: Runtime Byte Code, Blocks and Blockchain, Fee Schedule –Supporting Protocols –Solidity Language

Block 4: WEB3 AND HYPERLEDGER

Introduction to Web3 –Contract Deployment –POST Requests –Development Frameworks –Hyperledger as a Protocol –The Reference Architecture –Hyperledger Fabric –Distributed Ledger –Corda.

Block 5: ALTERNATIVE BLOCKCHAINS AND NEXT EMERGING TRENDS

Kadena –Ripple–Rootstock –Quorum –Tendermint –Scalability –Privacy –Other Challenges
–Blockchain Research –Notable Projects –Miscellaneous Tools.

Reference book:

1. Imran Bashir, “Mastering Blockchain: Distributed Ledger Technology, Decentralization and Smart Contracts Explained”, Second Edition, Packt Publishing, 2018.
2. Arshdeep Bahga, Vijay Madisetti, “Blockchain Applications: A Hands On Approach”, VPT, 2017.
3. Andreas Antonopoulos, Satoshi Nakamoto, “Mastering Bitcoin”, O’Reilly, 2014.
4. Roger Wattenhofer, “The Science of the Blockchain” CreateSpace Independent Publishing, 2016.
5. A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, “Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction”, Princeton University Press, 2016.
6. Alex Leverington, “Ethereum Programming” Packt Publishing, 2017



Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Software Project management (Elective II)
COURSE CODE : MSCS-22
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Software Project management, the student shall be able to:

- Understand the framework of project management
 - Learn to monitor and control the project
 - Know the sound knowledge in Agile method
 - Know the team, cost, quality and resource management
 - Identify and control the risk in the projects
-

COURSE OUTCOMES

After completion of the Software Project management, the student will be able to:

- Analyze the scope, cost, timing, and quality of the project, at all times focused on project success as defined by project stakeholders.
- Align the project to the organization's strategic plans and business justification throughout its lifecycle.
- Identify project goals, constraints, deliverables, performance criteria, control needs, and resource requirements in consultation with stakeholders.
- Implement project management knowledge, processes, lifecycle and the embodied concepts, tools and techniques in order to achieve project success.
- Adapt projects in response to issues that arise internally and externally.

Block 1:

Project Management Framework: Introduction: Project - Project management - Relationship among Project, Program and Portfolio management - Project and operations management- Role of project manager - Project management body of knowledge - Enterprise Environmental factors. Project life cycle and Organization: Overview of project life cycle - Projects vs Operational Work - Stakeholders - Organizational influences on project management. **The Standard for Project Management of a Project:** Project management processes for a project: Common project management process interactions - Projects management process groups - Initiating process group - planning process group - Executing process group - Monitoring and controlling process group - Closing process group.

Block 2:

Choosing Methodologies and Technologies – Software Processes and Process Models – Choice of Process Models – The Waterfall Model– Prototyping – other ways of categorizing prototype - **Agile Methods** – Extreme Programming Selecting the Most Appropriate Process Model- Need of Agile - Iterative vs Incremental-Agile Manifesto and Mindset – Lean, Scrum and Kanban methods-uncertainty, Risk, and lifecycle selection-Scrum Elements overview-5 levels of planning-Scrum Process overview-Agile Team-roles and responsibilities- Epic-feature-User Stories-PBI-The Sprint.

Block 3:

The Project Management Knowledge Areas: Project integration management: Develop project charter - Develop project management plan - Direct and manage project execution - Monitor and control project work - Perform integrated change control - Close project or phase. Project scope management: Collect requirements - Define Scope - Create WBS - Verify Scope - Control Scope. Project team management: Define activities - Sequence activities - Estimate activity resources - Estimate Activity Durations - Develop Schedule - Control Schedule.

Block 4:

Project cost management: Estimate costs - Determine budget - Control costs. Project Quality Management: Plan quality - perform quality assurance - Perform quality control. Project Human Resource Management: Develop human resource plan - Acquire project team - Develop project team - Manage project team. Project Communications Management: Identify stakeholders - Plan communications - Distribute information - Manage stakeholder expectations - report performance.

Block 5:

Project Risk Management: Plan risk management - Identify risks - Perform qualitative risk analysis - Perform quantitative risk analysis - plan risk responses - Monitor and control risks. Project Procurement Management: Plan - Conduct - Administer - Close procurements.

Reference Books:

1. "A guide to the Project management Body of Knowledge (PMBOK Guide)" Fourth Edition, Project Management Institute, Pennsylvania, 2008
2. BOB Huges, Mike Cotterell, Rajib Mall "Software Project Management", McGraw Hill, Fifth Edition, 2011.
3. Emerson, "Agile Handbook," Philosophie
1. Futrell, "Quality Software Project Management", Pearson Education India.
2. Royce, "Software Project Management", Pearson Education India.
3. C.Ravindranath Pandian, "Applied Software Risk Management-A Guide for Software Project Managers", Auerbach Publications, 2015.
4. Benjamin A. Lieberman, "The Art of Software Modeling", Auerbach Publications, 2010.



Tamil Nadu Open University
School of Computer Science
Chennai – 15

Mater of Science in Computer Science - Syllabus – III Semester (Distance Mode)

COURSE TITLE : Software Quality Assurance (Elective II)
COURSE CODE : MSCS-23
COURSE CREDIT : 04

COURSE OBJECTIVES

While studying the Software Quality Assurance, the student shall be able to:

- To find defects which may get created by the programmer while software development.
 - To gain confidence in and giving information about the quality level.
 - Ensuring that the end result meets the user and business requirements.
-

COURSE OUTCOMES

After completion of the Software Quality Assurance, the student will be able to:

- Get the Basics of software quality assurance
- Know the Introduction to software testing
- Get knowledge of Different testing techniques
- Understand the Automated testing concepts

Block 1:

Introduction: Concepts of Quality Control, Quality Assurance, Quality Management - Total Quality Management- Cost of Quality-QC tools - 7 QC Tools and Modern Tools- Other related topics - Business Process Re-engineering - Zero Defect, Six Sigma- Quality Function Deployment- Benchmarking- Statistical process control.

Block 2:

Software Quality Assurance Models: Software Quality Assurance; Statistical Quality Assurance - Software Reliability- Models for Quality Assurance - ISO-9000 – Series- CMM- SPICE.

Block 3:

Software Processes: Software Process - Definition and implementation- internal Auditing and Assessments.

Block 4:

Software testing: Concepts- Tools-Reviews-Inspections & Walkthroughs- P-CMM.

Block 5:

Total Quality Management (TQM): Introduction- Software reuse for TQM- Software testing method for TQM- Defect Prevention and Total Quality Management-Zero Defect Software Development-Clean room Engineering.

Reference Book:

1. Watt.S. Humphery, " Managing Software Process ", Addison - Wesley, 2008.
2. "Software quality Theory & Management ", Allan Gillies, Thomson international Press 1997. (Block 1 I & II)
3. "Software Engineering ", Roger Pressman, 5th edition McGraw Hill, 1999
4. "Total Quality Management for Software", G.Gordan Schulmeyer, James, International Thomson Computer Press, 1998
5. Bob Hughes, Mike Cotterell "Software Project Management" , Tata Mc Graw Hill 3rd Edition, 2010.